1995-03

Scheduling basic training for the Federal Armed Forces of Germany

Drews, Ralf.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/7529

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SCHEDULING BASIC TRAINING
FOR THE FEDERAL
ARMED FORCES OF GERMANY

by

Ralf Drews

March, 1995

Thesis Advisor: Gordon H. Bradley

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved OMB No.0704

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March, 1995	Master's Thesis

4. TITLE AND SUBTITLE SCHEDULING BASIC TRAINING FOR THE FEDERAL ARMED FORCES OF GERMANY

5. FUNDING NUMBERS

6. AUTHOR(S)
Drews, Ralf

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This thesis describes the design, development and testing of a personal computer-based system to schedule basic training for Federal Armed Forces of Germany (*Bundeswehr*) conscripts. Each quarter 20,000 conscripts undergo three months of basic training in over 100 different units. The training objectives fit in 450 training topics which are currently scheduled in a time consuming manual process. An interactive PASCAL program with a graphical user interface heuristically constructs a schedule and accommodates manual changes. The heuristic uses a clock advancing selection algorithm followed by an improvement algorithm. After pre-assigning topics that require a special facility or instructor, the heuristic schedules the remaining topics while satisfying all prerequisites. Extensive testing of the program on realistic data shows that the software produces a high quality schedule with face validity and requires less than ten minutes on an IBM 80486DX-2 66MHz personal computer to construct a schedule for one quarter. It is estimated that quarterly data entry, schedule review and revision with the program would require 1.5 man-days; this is a significant savings compared to the 10 to 15 man-days currently needed without the program. Considering 100 units must produce a quarterly schedule four times a year, the program would cut the *Bundeswehr* yearly scheduling effort from between 4,000 and 6,000 man-days to an estimated 600 man-days.

14. SUBJECT TERMS
Scheduling, Time Tables, Heuristic, Computer, Graphical User Interface, Manual Changes

15. NUMBER OF PAGES 124

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

SCHEDULING BASIC TRAINING FOR THE FEDERAL ARMED FORCES OF GERMANY

Ralf Drews
Captain, German Army
M.S., Federal Armed Forces University, Hamburg, 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1995

Author: [redacted]
Ralf Drews

Approved by: [redacted]
Gordon H. Bradley, Thesis Advisor

[redacted]
Robert F. Dell, Second Reader

[redacted]
Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

This thesis describes the design, development and testing of a personal computer-based system to schedule basic training for Federal Armed Forces of Germany (*Bundeswehr*) conscripts. Each quarter 20,000 conscripts undergo three months of basic training in over 100 different units. The training objectives fit in 450 training topics which are currently scheduled in a time consuming manual process. An interactive PASCAL program with a graphical user interface heuristically constructs a schedule and accommodates manual changes. The heuristic uses a clock advancing selection algorithm followed by an improvement algorithm. After pre-assigning topics that require a special facility or instructor, the heuristic schedules the remaining topics while satisfying all prerequisites. Extensive testing of the program on realistic data shows that the software produces a high quality schedule with face validity and requires less than ten minutes on an IBM 80486DX-2 66MHz personal computer to construct a schedule for one quarter. It is estimated that quarterly data entry, schedule review and revision with the program would require 1.5 man-days; this is a significant savings compared to the 10 to 15 man-days currently needed without the program. Considering 100 units must produce a quarterly schedule four times a year, the program would cut the *Bundeswehr* yearly scheduling effort from between 4,000 and 6,000 man-days to an estimated 600 man-days.

THESIS DISCLAIMER

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logical errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

The Germany Army (*Bundeswehr*) is a conscript force that drafts approximately 20,000 people quarterly to serve for 12 months. During the first three months draftees undergo basic training in special basic training units and must satisfy more than 450 training objectives. A Master Sergeant or Chief Master Sergeant currently schedules the 450 training objectives for a company of these conscripts (approximately 150 soldiers). It takes the scheduler approximately two to three weeks each quarter to manually perform this complicated duty. This time is valuable and keeps the soldier from pursuing his other duties. For this reason a complete quarterly schedule is seldom developed. For most quarters the first three to four weeks are completed and the remaining scheduling is performed on a week to week basis. This thesis develops a computer program to quickly and efficiently produce a quarterly schedule.

Producing a quarterly schedule is a complicated and time-consuming task since it must account for a number of constraints. Many training objectives have inter-dependencies or requirements for special instructors and facilities. These special instructors (Security Personnel, Medical Personnel, Military Priest, and Social Worker) and facilities (Garrison Training Area, Firing Range, Gymnasium, and Obstacle Course) are scarce resources and not always available. The schedule should also have an appropriate mix of lectures and physical activities. Legal and clerical holidays as well as possible days of leave also increase the complexity of the schedule.

The program with a graphical user interface constructs a schedule using decision rules similar to those the scheduler employs and accommodates manual changes. After pre-assigning topics that require a special facility or instructor, the program schedules the remaining topics while satisfying all prerequisites. Before presenting the result to the scheduler for evaluation the program tries to improve its solution by testing local interchanges of assignments.

The user can modify the computer calculated solution with additional help from the machine which tests whether requests for changes conform to the requirements and interdependencies. If the program detects possible violations in the user request it alerts the user and requires a confirmation before implementing the change.

Extensive testing of the program on realistic data shows that the software produces a high quality schedule with face validity and requires less than ten minutes on an IBM 80486DX-2 66MHz personal computer to construct a schedule for one quarter.

It is estimated that quarterly data entry, schedule review and revision with the program would require 1.5 man-days; this is a significant savings compared to the 10 to 15 man-days currently needed without the program. Considering 100 units must produce a quarterly schedule four times a year, the program would cut the *Bundeswehr* yearly scheduling effort from between 4,000 and 6,000 man-days to an estimated 600 man-days.

I. INTRODUCTION

The Germany Army (*Bundeswehr*) is a conscript force that drafts approximately 20,000 people quarterly to serve for 12 months. During the first three months draftees undergo basic training in special basic training units and must satisfy more than 450 training objectives. A Master Sergeant or Chief Master Sergeant currently schedules the 450 training objectives for a company of these conscripts (approximately 150 soldiers). It takes the scheduler approximately two to three weeks each quarter to manually perform this complicated duty. This time is valuable and keeps the soldier from pursuing his other duties. This thesis develops a computer program to quickly and efficiently produce a quarterly schedule.

A. BACKGROUND

The problem consists of assigning topics that comprise more than 450 objectives to available time periods and in cases when all topics cannot be scheduled, deciding which topics to leave out of the schedule. Each training objective is achieved in one or two time periods. The length of a single period is 45 minutes from 0700 until 1700 hours and 60 minutes at all other times. Interdependencies among the training topics and prerequisites among them increase the difficulty of scheduling.

The training objectives can be divided into four groups. The first consists of a large track where each topic builds on previous topics and interchanges in the sequence are nearly impossible. The second group (approximately 25 percent) has training objectives with up to ten other topics as prerequisites. The next group (30 percent) has up to four or five prerequisites and the fourth and last group (40 percent) has no prerequisites.

The training objectives are specified in the Catalog of Training Objectives (CTO) (*Heeresamt*, 1988). This catalog structures the topics into main groups that are named according to a joint forces naming convention. A total of 14 different topic groups are employed in basic training. Each group consists of multiple topics and can be identified by a two-digit code. These groups, the number of topics and the time requirements are shown in Table 1. The groups are not independent and a topic may have prerequisites in other main groups; also topics from different groups may require the same training facilities. To guide manual scheduling and enforce at least to some degree conformity in training over all

basic training units, the CTO provides a flowchart that outlines suggested scheduling of topic groups or subgroups into sequences and also indicates prerequisite chains (*Heeresamt*, 1988, Anlage 1). It does not provide a fixed requirement in weeks and does not have a time scale.

Topic Main Group	Main Code	Number of Entries	Hours
General Combat Training For All Branches	01	88	135
Anti Tank Warfare / Identification	02	27	22
Engineer Service	03	2	4
NBC Defense And Self Protection	04	8	13
Basic Medical Training	06	15	22
Training With Weaponry And Material of All Branches	08	3	4
Gunnery Training With Handheld Weapons And Live Firing	09	35	70
Sports	10	26	31
Formation Training	11	20	12
General Knowledge About Troops And Organization	14	3	3
Military Cure Of Souls And Character Guidance	15	3	6
Civic Education And Leadership	16	23	29
Military Security / Military Intelligence	20	13	21
General Services	99	145	150
TOTAL:		411	522

Table 1: Topics Breakdown in Main Groups in the Catalog of Training Objectives

Anti Tank Warfare / Identification and Formation Training, shown in Table 1, have more entries than hours to be scheduled because not all topics in these groups need to be done. The company commander chooses topics in these groups according to his available assets.

Certain topics have to be done at a special facility and nowhere else. These facilities are usual key facilities that have limited availability since they have to be shared with other military units. This complicates the scheduling process on one hand but on the other hand gives it a certain structure. Five key training facilities are identified in the CTO; the other training can be done at company owned facilities. The key facilities with their codes (indices) used in the computer program described in Chapter III are shown in Table 2.

Most of the training is done under the command and control of the platoon leader which allows for all platoons to have most objectives scheduled simultaneously. Nearly 20

percent of the training, however, has to be done by the Company Commander or the Master Sergeant of the Company which leads to scheduling conflicts because they can only train one platoon at a time. Nonavailability of instructors in certain periods due to other commitments increases the complexity of the scheduling problem. Other topics have to be given by special instructors who are not company personnel. This requires coordination. The instructors identified in the CTO are shown in Table 3. For later use the abbreviation and the computer code is included.

Facility Type	Code Number
Garrison Training Area	1
Firing Range	2
Gymnasium / Athletic Field	3
(Outdoor/Indoor) Swimming Pool	4
Obstacle Course	5

Table 2: Key Training Facilities in CTO and Their Computer Representation

Position	Abbreviation	Code Number
Battalion Commander	BtlCdr	0
Company Commander	CpCdr	1
Chief Master Sergeant	CMSgt	2
Platoon Leader	PltLdr	3
NBC-Defense Sergeant	NBCSgt	4
Military Priest	MilPriest	5
Medical Personal	MedPers	6
Social Worker	SocWork	7
S2 Personal	SecPers	8

Table 3: Instructor Personnel with Abbreviations and Computer Representation

Additional lower priority restrictions, such as providing an appropriate daily mix of theoretical and practical instructions and ensuring special physical training does not happen directly after a meal break, have to be incorporated as well. To facilitate the observation of those restrictions each topic is given an identifier that specifies whether the topic is a practical (manual) exercise or a lecture.

The whole schedule is currently produced by hand in a labor and time intensive process. After the external training facilities have been assigned, the company commander gives a rough outline of his intentions to the scheduler, who then sits down and tries to incorporate all requirements and restrictions into a feasible schedule for the platoons.

After producing what he thinks is a feasible schedule, the result is reviewed by the company commander and the platoon leaders. If they find topics they think could be taught better at different periods, the Sergeant tries to incorporate these changes. Due to the complexity of the scheduling problem, incorporating changes while ensuring all restrictions are observed can be a time consuming process.

B. RELATED WORK

Models for scheduling or timetabling that appear in the Operations Research literature often deal with special situations and are tailored for the needs of the special user. The most similar scheduling models are school or classroom models (Schmidt, 1979). Those models focusing on classroom or course scheduling deal with only a moderate number of different topics and focus on one week. The other weeks in a course sequence are always assumed to be the same (de Werra, 1985 or Mulvey, 1982). The main emphasis in those models is to find a course scheduling combination that allows most students to take their favorite combination and minimize the number of students who have to switch courses. These models employ either linear programming techniques (Lawrie, 1969) or graph coloring (Wood, 1969).

Another focus of available models is examination scheduling. Those algorithms try to schedule exams such that no student has to take two in one day and other criterion are met. Other criterion are often specified by the policies of the institution that uses the model. Those models also have limited scope and deal with only a small number of available time periods to schedule the exams, usually a week or at most two weeks. (Oakford, 1966 and Wood, 1968)

Justice (1993) looks at scheduling class sequences needed by soldiers in the U.S. Marine Corps to qualify for an electronics communication specialty. The focus of his scheduling problem was minimizing the time between different classes rather than scheduling individual topics in the courses (the focus of this thesis).

None of the models published so far deals with a whole quarter with distinct topics to be scheduled each day. On the other hand, trainees do not have individual schedules; the trainees in each platoon have the same track. The complexity and difficulty of scheduling is due to the large number of topics, the large number of possible assignments, and ensuring the requirements and prerequisites are satisfied.

Most of the models found in literature were not designed for personal computers. Most models were written prior to the advent of personal computing and were therefore designed to execute on mainframe computers (as Leighton, 1979 and Romero, 1982). The model developed here has to be used in the company that can only employ programs that run on personal computers.

C. HARDWARE AND SOFTWARE CONSIDERATIONS

The primary goal is to develop a program that can solve the problem in a reasonable time to convert a labor intensive manual task into a computer assisted one. Secondly, it should work with the available hardware and software, as described below. If a high quality solution cannot be achieved with existing hardware and software, this requirement will be superseded by the primary goal. As the solution will show, the problem can be solved on the existing equipment.

1. Hardware Availability

The available hardware is an IBM-compatible system with a 80386SX processor and at most four Megabyte of random access memory (RAM). The system is equipped with a fixed disk where the capacity depends on the date the system was fielded. The low end systems have a storage capacity of only 40 Megabytes. The systems are distributed as stand-alone systems. They have not and will not have in the near future a modem or access via a modem to a central computing facility. The attached printer is a dot-matrix printer, either 9-pin or 24-pin, with a narrow carriage. Therefore, only printing in portrait orientation is possible.

2. Software Availability

The given systems are DOS based using the Microsoft™ DOS 5.0 version of the disk operating system. No systems are equipped with Windows and for the near future the fielding of Windows for all systems cannot be expected. They come equipped with a wordprocessor, a DOS database and a small drawing program. No programming languages except the supplied Basic are available and there are no other application programs. The main use of the machine is that of a typewriter with storage capability.

D. USER PROFILE

The main user for this scheduling program is a Master Sergeant or Chief Master Sergeant who has joined the armed services either as a long term volunteer (12 to 15 years) or as a professional. His past experiences is ideally that of a platoon leader although occasionally Sergeants without this experience get assigned to the position of Company Squad Leader and Nuclear - Biological - Chemical - Defense Sergeant (NBC-Defense Sgt.). The average time he spends on the job ranges form three to five years.

Computer knowledge is not a prerequisite for getting this assignment. In combat and combat support branches, the computer is not widely used and thus there is no need to train every soldier how to use one. Therefore, the new scheduler could have had no exposure to computers, either in the military or civilian life.

Because of the potential lack of users' exposure to computers, the scheduling program must be easy to use; that is, it should have a low learning curve. The program should also have a sufficient amount of internal error checking to support the unsophisticated user and the novice.

E. DATA

The data for the schedule and thus for the program can be classified into three groups:

- unchanging data such as in the catalog of training objectives;
- sometimes changing data such as the basic weekly schedule and the number of platoons per quarter;
- quarterly changing data such as legal and clerical holidays, key facility assignments, and instructor availability.

The training objectives in the CTO can be considered unchanging data. They could be supplied with the distribution of the program. In the rare event the CTO or parts of it change those changes should be maintained centrally. They should be handled on the same level as changes in regulations. The distribution of this data set with the program and subsequent updates in the form of normal regulation maintenance and changing procedures is the safest way to keep every single copy at the same level. Using this approach reduces the computer knowledge a scheduler has to have. Thus the scheduler has no need to bother with the way files are stored and in what format they might be saved.

Some data that are not likely to change often are shared between quarters. Although the program sets them automatically, at the beginning of the work on a new quarter the scheduler may display and change them. This reduces the data input in subsequent uses of the program and should not interfere with the work. Although two quarters can be handled simultaneously, the recalculation of the current quarter at a time when the data for the next quarter are all available is unlikely. Usually the data are available three to four weeks prior to the beginning of the quarter. Therefore, the recalculation of the current quarter does not provide a meaningful result because large portions have already been executed and the schedules cannot be changed any more.

All the other data change every quarter. The legal and clerical holidays are not built into the basic calendar. The regulations differ for all states in the Federal Republic of Germany and this piece of information can be given by the user easily and quickly. It requires filling only one dialog box. Additionally, this provides the opportunity to supply data in one input step about other days of leave, special leave or leave for additional duty.

Most of the other data has to be supplied by the user. Due to the diversity of garrison structures and regulations it is not possible to provide a basic one-fits-all data set. Following the idea of mission-type tactics, the company commander needs the freedom to make basic training decisions and use the program as a decision aid and not as a controller and regulator.

II. MATHEMATICAL DESCRIPTION

This chapter develops a mathematical description of the problem outlined in Chapter I. The description contains both the data structure used to implement the heuristic as well as an integer programming formulation.

Measures of effectiveness are defined; they are used to evaluate the goodness of the schedules that are constructed.

A. MEASURES OF EFFECTIVENESS

Any schedule that is produced by a person or by a machine has to be evaluated for quality. To make comparisons between results, common quantified measures have to be applied. The users' subjective feelings about a schedule cannot be easily taken into account in evaluating a computer constructed solution. A single measure of effectiveness (MOE) for the whole schedule does not seem appropriate due to the complexity of the task.

Some MOEs picked to measure the quality of a solution are simple counts. Other MOEs involve more computation such as the penalty for a topic that could not be scheduled being a unitless measure of the relative importance of a topic. Another MOE uses the distances in periods from the CTO suggested time frame for a topic and the actual period it is scheduled. The following summarizes the MOEs:

- Sum of all penalties for topics that could not be scheduled (MOE1);
- Number of topics that could not be scheduled;
- Sum of the distances between the suggested periods for scheduling and the actual period a topic is scheduled:
  - for topics that do not need a special facility or external personnel (MOE2),
  - for topics on a Garrison Training Area (MOE2g),
  - for topics on a Firing Range (MOE2f),
  - for topics given by external instructors (MOE2o);
- Number of topics scheduled outside the suggested periods for scheduling:
  - for topics that do not need a special facility or external personnel (MOE2),
  - for topics on a Garrison Training Area (MOE2g),
  - for topics on a Firing Range (MOE2f),
  - for topics given by external instructors (MOE2o);

- Number of occurrences when three topics of type lecture are consecutively scheduled for the platoon (MOE3).

The abbreviations in parenthesis are used in the program and are also featured in reports given by the program (Appendix C). The MOE's are incorporated into the following mathematical description of the problem.

B. MODEL

The model description reflects data structures and solution mechanisms used. All data used in the integer program are marked with a vertical bar (|) on the left side. Other data are necessary for data preparation and output.

<u>Indices:</u>

|	t :	topics	(t = 1, 2, ..., MT),	currently MT = 450;
|	i :	instructors	(i = 1, 2, ..., NI),	currently NI = 8;
|	f :	facilities	(f = 1, 2, ..., NF),	currently NF = 5;
|	p :	periods	(p = 1, 2, ..., MP),	currently MP = 600;
|	pl:	platoons	(pl = 1, 2, ..., NP),	currently NP = 3;
	s :	Saturdays	1, 2, ..., 5; and	
	h :	holidays	1, 2, ..., 15.	

<u>Given Data:</u>

TL[t] : data structure that contains the CTO's list of topics.
Each entry in the structure consists of the following information:

	TL[t].CodeNumber	nine digit numeric code for the topic,
	TL[t].RunIndex	two digit running index of topics with the same numeric code,
|	TL[t].InstructionType	one character for classification of lecture or practical training (U, P), the type "U" for lecture results from the German "*Unterricht*" = "lecture",
|	TL[t].Instructor	instructor index (possible values are 1, 2, ..., or NI), as given in Table 3,
|	TL[t].Duration	length of time to reach the training objective (1, 2, ..., or 7),
	TL[t].DayTime	classification of daytime objective or objective to be taught under conditions of limited visibility (D, N),

TL[t].Facility	facility type (possible values are 1, 2, ..., or NF), as in Table 2,
TL[t].SubFacility	classification of range length for Firing Ranges (A, B, C, D),
TL[t].FixedTime	classification whether the topic has been prescheduled at a fixed time by user,
TL[t].Prerequisite	linked list of numeric codes specifying prerequisites for this topic,
TL[t].Penalty	integer value for not using this topic in scheduling (1, 2, ..., 1000),
TL[t].Earliest	number of the earliest week topic t can be scheduled, (from the flowchart in the catalog of training objectives),
TL[t].Latest	number of the latest week topic t can be scheduled, (from the flowchart in the catalog of training objectives),
TL[t].Scheduled[pl]	Boolean entry for each platoon in an array, 0 if topic has not been scheduled, 1 otherwise.

BS[p] : basic weekly schedule for the company with start time and end time for each period of the week (p = 1, 2, ..., 46),

F[f] : assignment of facility f for the company, supplied by user and given in actual calendar dates and clock times (f = 1, 2, ..., NF),

I[i] : availability of outside instructors and unavailability of company instructors given in actual calendar dates and clock times (I = 1, 2,..., NI),

C : main calendar data for the quarter,

C.Arrival	date of arrival for new conscripts,
C.Departure	date of departure for then trained conscripts,
C.Sat[s]	dates for Saturdays and Sundays with duty (s = 1, 2, ..., 5),
C.Hol[h]	dates with legal and clerical holidays in the quarter as well as special leave dates (h = 1, 2, ..., 15)

R : requirements from division or battalion level for topics in the catalog given in the format:

R.TopicCode	the nine digit codenumber of the topic,
R.Req	the requirement for this topic, i.e. : when to teach, when not to teach, how often per week

S : special topics that are not in the catalog of training objectives, given as:

S.Date	the date this topic has to take place,
S.Time	the time of the day,
S.Instructor	the instructor who has to instruct this topic,
S.Title	the title of the topic.

Derived Data:

P[p] : total enumeration of all periods in the quarter from given data in BS[p] and F[f], thus only those periods that can be scheduled are enumerated; in supplement to the index naming the period the following information is kept (p = 1, ..., MP):

P[p].Dates	the calendar date of period p,
P[p].Times	the time frame of period p,
P[p].DayTime	the daytime of the period, with differences between periods in Summer or Winter half of the year,
P[p].DayOfWeek	the day of the week for period p.

Fac[f] : period equivalent coding of data supplied by user in F[f], with the following structure (f = 1, 2, ..., NF):

Fac[f].SubFacility	range length coding for Firing Ranges (A, B, C, D),
Fac[f].Period	the periods the facility is available,
Fac[f].Next	pointer to the next available period for facility f.

{FCapacity[f, p] : For the integer program that follows, this is the number of platoons that can train in facility f in period p. This can be formed from the Fac[f] data structure.}

Ins[i] : period equivalent coding for the data about instructors supplied by user in I[i] (I = 1, 2, ..., NI):

Ins[i].Period	the period instructor i is available, if outside instructor, or not available otherwise,
Ins[i].Next	pointer to the next period information for instructor i.

{ICapacity[i, p] : For the integer program that follows, this is the number of platoons that can be trained by instructor i in period p. This can be formed from the Ins[i] data structure.}

RL : requirement list coding from data R into period notation with the format:

RL.TopicCode	the nine digit numeric code of the topic,

RL.OnlyAtPeriod	period number, if requirement that this topic should be given only at certain periods,
RL.NotAtPeriod	period number, if requirement that this topic should not to be given at certain periods,
RL.NoMoreThan	a single number, if the topic should not be given more than that many times per week,
RL.AtLeast	a single number, if the topic should be given at least that many times per week.

E[pl, t] : earliest starting period for topic t, derived from TL[t].Earliest (pl = 1, ..., NP; t = 1, ..., MT).

L[pl, t] : latest starting period for topic t, derived from TL[t].Latest (pl = 1, ..., NP; t = 1, ..., MT).

Decision Variable:

Sched[t, pl, p] : 1 if topic t is scheduled to start for platoon pl at period p, 0 otherwise, (t = 1, ..., MT; pl = 1, ..., NP; p = 1, ..., MP)

Z[pl, p] : 1 if period p, p + 1 and p + 2 all have a lecture topic for platoon pl, 0 otherwise, (pl = 1, ..., NP; p = 1, ..., MP)

Formulation:

$$\text{Minimize} \quad \sum_{pl}\sum_{t} TL[t].Penalty * \left(1 - \sum_{p} Sched[t,pl,p]\right) \tag{1}$$

$$\text{Minimize} \quad \sum_{pl}\sum_{t} \left(1 - \sum_{p} Sched[t,pl,p]\right) \tag{2}$$

$$\begin{aligned}\text{Minimize} \quad & \sum_{t}\sum_{pl}\sum_{p<E[pl,t]} (E[pl,t] - p) * Sched[t,pl,p] + \\ & \sum_{t}\sum_{pl}\sum_{p>L[pl,t]} (p - L[pl,t]) * Sched[t,pl,p]\end{aligned} \tag{3}$$

$$\text{Minimize} \quad \sum_{pl}\sum_{t}\sum_{p<E[pl,t]} Sched[t,pl,p] + \sum_{pl}\sum_{t}\sum_{p>L[pl,t]} Sched[t,pl,p] \tag{4}$$

$$\text{Minimize} \quad \sum_{pl}\sum_{p} Z[pl,p] \tag{5}$$

subject to:

$$\sum_{p'=p-TL[t].Duration+1}^{p} \; \sum_{pl} Sched[t,pl,p'] \le 1$$

$$\forall\ p = 1, \ldots, MP;\ t \text{ such that } TL[t].Instructor \neq PltLdr \tag{6}$$

$$\sum_{t} \sum_{p'=p-TL[t].Duration+1}^{p} Sched[t,pl,p'] = 1 \qquad \forall pl = 1, \ldots NP;\ p = 1, \ldots MP \tag{7}$$

$$\sum_{t \text{ s.t. } TL[t].Facility=f} \sum_{pl} \sum_{p'=p-TL[t].Duration+1}^{p} Sched[t,pl,p'] \le FCapacity[f,p]$$

$$\forall\ p = 1, \ldots, MP;\ f = 1, \ldots, NF \tag{8}$$

$$\begin{aligned}
&\sum_{t \text{ s.t. } TL[t].InstructType=U} Sched[t,pl,p-1]*(TL[t].Duration-1)+ \\
&\sum_{t \text{ s.t. } TL[t].InstructType=U} Sched[t,pl,p]*TL[t].Duration+ \\
&\sum_{t \text{ s.t. } TL[t].InstructType=U} Sched[t,pl,p+1]*TL[t].Duration+ \\
&\sum_{t \text{ s.t. } TL[t].InstructType=U} Sched[t,pl,p+2] \le (2+Z[pl,p])
\end{aligned}$$

$$\forall pl = 1, \ldots NP;\ p = 1, \ldots MP\text{-}2 \tag{9}$$

$$\sum_{t \text{ s.t. } TL[t].Instructor=i} \sum_{pl} \sum_{p'=p-TL[t].Duration+1}^{p} Sched[t,pl,p'] \le ICapacity[i,p]$$

$$\forall\ p = 1, \ldots, MP;\ I = 1, \ldots, NI \tag{10}$$

$$Sched[t,pl,p] \le \sum_{p'<p-TL[t'].Duration+1} Sched[t',pl,p']$$

$$\forall\ p = 2, 3, \ldots, MP;\ pl = 1, \ldots, NP;\ (t,t') \text{ s.t. } t' \in TL[t].Prerequisite \tag{11}$$

The first objective function, Equation (1), corresponds to MOE1 and sums the penalties for the topics that could not be scheduled. Equation (2) counts the total number of topics not scheduled. The Equation (3) sums the absolute distances between the actual period a topic is assigned to and the closest interval limit of the period numbers derived from the flowchart in the CTO. Values in the interval incur no weight in this objective function. This objective function corresponds to MOE2 and its subdivisions. Equation (4) sums the actual number of topics scheduled outside the suggested timeframe. The last objective function (5) counts the number of occurrences when three or more topics of lecture type are scheduled back to back.

The first constraint, Equation (6), allows each topic that requires an instructor who is not the platoon leader to be scheduled only for a single platoon in a period. The next constraint (7) ensures scheduling for exactly one topic in each period for a single platoon. Equation (8) enforces that the facility required for a topic is available when the topic is scheduled and allows only as many platoons on the facility as capacity available. The forth constraint (9) sets the value in the array Z[pl, p] to 1 if three topics of the same type are scheduled back to back. If more than three topics of lecture type are scheduled consecutively, for example in the periods p, p + 1, p + 2 and p + 3, then the entries in the array for Z[pl, p] and Z[pl, p + 1] must both equal 1. The next constraint (10) enforces that an instructor is only scheduled if available. The last constraint (11) insures that each topic is assigned in the schedule only if the prerequisites have been assigned prior to the period the topic has to start.

In the decision variable Sched[t, pl, p] all those entries that correspond to combinations of t and p that can never be used are set to zero in the initialization. For example, these are the combinations where the day / night identifier for the topic and the period do not match. Thus saying, TL[t].DayTime ≠ Period[p].DayTime.

This integer program would have approximately the following number of constraints:

- for (6): 600*450 = 270,000 equations in the worst case (approximately 40 percent of these constraints would be needed in the test problems considered);
- for (7): 3*600 = 1,800 equations;
- for (8): 5*600 = 3,000 equations;
- for (9): 3*598 = 1,794 equations;
- for (10): 8*600 = 4,800 equations;
- for (11): 3*599*225*450 = 181,946,250 equations in the worst case (approximately five percent of these constraints would be needed in the test problems considered);

This then would amount to a total of approximately 9,216,707 equations for the constraints. The number could go up in the worst case, considering all totals, to 182,227,644 equations. In the worst case, the Sched[t, pl, p] would have 810,000 (450*3*600) binary variables, and 1,800 (3*600) Z[pl, p] binary variables.

It is possible to give a description with fewer variables. The schedule array Sched[t, pl, p] should be converted from a zero-one array to an integer array using the actual topic index as an entry. This reduces the storage needs of the problem.

To illustrate the connection between the two models the three-dimensional array above is mapped into the two-dimensional array below using the following mapping function:

$$Sched[pl,p] = t \text{ if and only if } \sum_{p'=p-TL[t].Duration+1}^{p} Sched[t,pl,p'] = 1$$

$$\forall \text{ pl} = 1, \ldots, \text{NP}; \text{ p} = 1, \ldots, \text{MP} \qquad (12)$$

The alternate mathematical formulation which is closer to the actual computer evaluation illustrates how the MOE's are calculated using the two-dimensional array. Those MOE's correspond to the objective functions from the mathematical description.

Formulation:

Minimize

$$\sum_{pl}\sum_{p} TL[Sched[pl,p]].Penalty*(1-TL[Sched[pl,p]].Scheduled[pl]) \qquad (12)$$

$$\text{Minimize } \sum_{pl}\sum_{p}(1-TL[Sched[pl,p]].Scheduled[pl]) \qquad (13)$$

$$\text{Minimize } \sum_{pl}\sum_{p} \begin{matrix} (E[pl,Sched[pl,p]]-p) \ \ if\ (p \le E[pl,Sched[pl,p]]) + \\ (p-L[pl,Sched[pl,p]]) \ \ if\ (L[pl,Sched[pl,p]] \le p) \end{matrix} \qquad (14)$$

$$\text{Minimize } \sum_{pl}\sum_{p} 1\ if\ (p \le E[pl,Sched[pl,p]]) + 1\ if\ (L[pl,Sched[pl,p]] \le p) \qquad (15)$$

$$\text{Minimize } \sum_{pl}\sum_{p} Z[pl,p] \qquad (16)$$

III. SOLUTION APPROACH

The integer program defined in the previous chapter would be difficult to solve exactly, particularly with existing hardware. Since it is highly unlikely that an exact solution could be achieved in a reasonable time, this thesis develops a heuristic to solve the scheduling problem.

This section describes the specially designed heuristic and then all related programming steps. It also outlines the steps used test the program. In the chapter a distinction is made between program, algorithm and heuristic. Program refers to the whole implementation including graphical user interface. Algorithm refers only to the actual calculation of the schedule and its related steps. Heuristic refers exclusively to the actual calculation of the schedule.

A. HEURISTIC

The heuristic is a greedy assignment heuristic. It calculates a set of eligible topics and sequentially picks assignments for available schedule periods. The heuristic can use many different selection and assignment rules. (This chapter presents a variety of alternate rules which are compared in Chapter V.)

Regardless of the selection rule employed, the heuristic first pre-assigns the two major training facilities (Garrison Training Area and Firing Range). It then sequentially assigns with limited backtracking from the set of eligible topics. When performing these assignments, the heuristic uses a period clock to determine of the next timeslot to assign. (A period clock refers to the method of starting with period one and after assigning to it advancing the clock to the next free (unassigned) period.)

For a given period, the set of eligible topics is the set of topics that can be scheduled. Their eligibility depends on their duration and prerequisites. To facilitate the calculation of the eligible set two additional sets of data are kept by determining for each unscheduled topic the earliest period and the latest period possible for scheduling. These additional sets of data take into consideration the already scheduled topics, necessity to consecutively schedule topic, as well as prerequisites.

One important factor in the assignment of topics to periods is the consideration of topics that are presented by an instructor who has to lecture each platoon. Therefore, the

eligible set for each platoon may be different and may depend on previously assigned topics for other platoons. If the instructor is already assigned to a platoon, a topic he has to lecture cannot be considered for scheduling at the same time for another platoon. For some periods there may not even be an eligible set because the period has already been scheduled. This condition applies to about 15 percent of all topics. Another factor making eligible sets different may be usage of the two pre-assigned training facilities.

A topic may also temporary leave the eligible set without being assigned since the algorithm does not allow topics whose duration is longer than the remaining periods of a day to be broken into two parts. Those topics then have to leave the eligible set temporarily and return if possible on the next day. The same construct applies to topics that are linked to restricted scheduling because of the availability of the required instructor or training facility. Those can only be in the eligible set if periods for scheduling are considered in which the instructor or training facility is available.

The following approach is then taken

Generic heuristic:

```
Initialize current period
Initialize the eligible set with all topics that can be scheduled in
this period
while there are free periods do
    select a topic from the eligible set and schedule it
    update the data sets for earliest and latest scheduling
    update the eligible set
    while a topic cannot be scheduled because earliest > latest
      and the topic change has not been considered before do
          backtrack by unscheduling topics and inserting the
          problematic topic at the first feasible period
```

Heuristic details follow:

1. The heuristic selects a topic out of the eligible set as either
    a. the one with the least freedom to be scheduled,
    b. the one with the highest penalty for not being used, or
    c. the one with the longest duration first.
2. The heuristic assigns the topic to either:
    a. earliest possible period,
    b. latest possible period, or
    c. some intermediate period.

3. The heuristic detects and breaks possible loops in backtracking by either
    a. allowing for at most one unscheduling and rescheduling of a topic,
    b. allowing rescheduling of an unscheduled topic only if the set of eligible topics has changed, not only because this topic reentered the set, but also due to additional topics entering or leaving, or
    c. taking at most n steps in backtracking.
4. The rules to break ties can be to either
    a. choose one at random, or
    b. an alternate selection rule.

As a result of the clock advancing heuristic, the period to be scheduled can be different for each platoon due to topics of different duration that have been assigned. To prevent drifting apart between the platoons, the platoon that is the furthest behind in scheduling, that is, whose period clock has the least period number, is considered next for scheduling. In case of ties the platoons are scheduled in numerical order. Due to this way of selecting the platoon in cases of ties the platoons with lower numerical identification might get better schedules and the last platoon worse schedules. To compensate for this, the schedules are interchanged between platoons where the past history of interchanges is taken into account to insure a fair distribution of schedule quality.

The value of the penalty for the unscheduled topics can be either static with the use of predefined penalties as part of the data or dynamic. The dynamic data are calculated from the given value in the data and the difference between the earliest and the latest possible period for scheduling. This takes into account the increasing difficulty with decreasing possibilities and so should reduce the amount of necessary backtracking.

After a schedule has been produced, an exchange algorithm sweeps the whole schedule and tests possibilities to improve the schedule by exchanging topics with each other. Specific improvement targets are to increase dispersion of topics from the same topic group, and to increase the mix between lectures and practical instruction.

B. ALGORITHM DESCRIPTION

The following section describes the actual implementation of the algorithm. The steps the algorithm takes in its decision process are stated explicitly and in depth.

1. Main Outline

The actual decision algorithm is embedded in a sequence of functions and procedures. The basic steps of the algorithm are:

- Convert the data into computer format;
- Validate the data;
- Preschedule major training facilities;
- Preschedule sports facilities and external instructors;
- Schedule remaining topics (main algorithm);
- Run improvement algorithm;
- Construct and display readable schedule.

The sequence of steps is also illustrated in the flowchart of Figure 1a and 1b. Some of the entries in the algorithm are explained here, others in the following chapters. During the whole calculation of the schedule the program displays a message box indicating that it is calculating the schedule.

After determining whether the user has supplied all necessary data to calculate the schedule, the conversion parts of the program as described in Appendix I are executed. After finishing this conversion, the topics that have to be performed on a Garrison Training Area are scheduled as described in detail in Chapter III.B.2.a. Intermediate statistics are written to the report file listing the number of topics assigned and the percentage of periods scheduled. The next step is the prescheduling of those topics to be done on Firing Ranges as explained in Chapter III.B.2.b. After these assignments are made additional statistics are calculated, this time explicitly for each platoon with the calculation of the percentage of topics scheduled for each platoon.

Because of the importance of physical fitness in the military the next assignment deals with the sports facilities. Gymnasium and Swimming Pool assignments are made by taking all available periods on a facility and only considering those that remain on days where neither a Garrison Training Area nor a Firing Range is available for a given platoon. Figure 2 shows the decision process for sports facility scheduling.

The algorithm chooses the platoon that has the least value for the index of sports topics in the list of training objectives and has not been scheduled on the day that is being considered. Then the topics in the list are assigned to have at most two consecutive periods (hours) of sports for this platoon. An additional consideration that they do not span over a lunch break is taken into account. The assigned topics are then flagged as scheduled and the index is advanced to the next unscheduled sports topic. Should there be more sports

facilities available than distinct sports topics required to schedule, the remaining sports facilities are used by repeatedly scheduling the last of the sports topics. This does not create a higher emphasis on those topics because most of the topics have the same objective.

In the next step of the algorithm the external instructors are scheduled. They are easy to schedule because they are usually only available for at most the number of periods they have to lecture. Therefore, the topics they have to lecture are assigned to those periods of availability. This procedure is shown in a flowchart in Figure 3. Each external instructor is considered individually because there is no possible interchange of topics among them. Those instructors who lecture each platoon individually are scheduled following the same approach used for sports facilities. This should maintain approximately the same level of training progress among the platoons for the topics each individual external instructor has to lecture. It should avoid having one platoon have all topics with an external instructor prior to any other platoon starting the training with this particular instructor. Because topics given by external instructors do not depend or build on each other the algorithm tests whether it can interchange the topic assignments for an instructor for a given platoon and thus improve the measures of effectiveness. After performing this scheduling, intermediate results on measures of effectiveness are written to the report file.

This completes the scheduling for the topics that require a key facility or an external instructor. The remaining topics are scheduled using the clock advancing algorithm. This algorithm is described separately in Chapter III.B.3. Having finished the schedule, the program tries to improve on that solution by employing an exchange algorithm as explained in Chapter III.B.4.

After finishing those calculations the final measures of effectiveness are written to the report file together with a listing of unscheduled topics for each platoon. The actual schedule is calculated by converting period data back into the normal date and time representation and constructing a display in a format familiar to the user. On each possible date and time the code number of the topic is listed. The actual computer representation of all data and the schedule is saved in a special log file to facilitate later revisions and manual changes of the schedule.

The last step of the algorithm is to display the final result beginning with the schedule for the first platoon in week one.

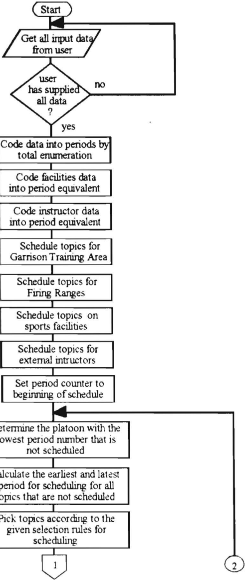

Figure 1a: Flowchart of Complete Scheduling Algorithm

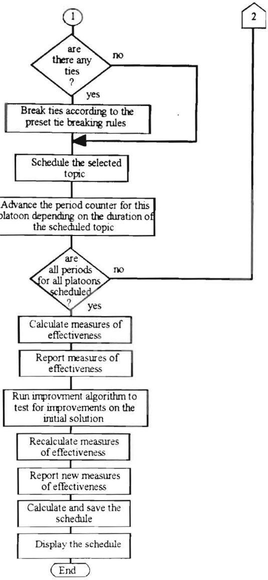

Figure 1b: Flowchart of Complete Scheduling Algorithm (Continued)

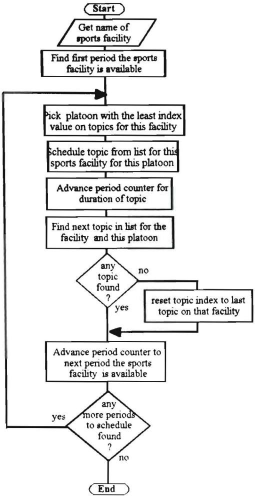

Figure 2: Flowchart for Scheduling Algorithm for Sports Facilities

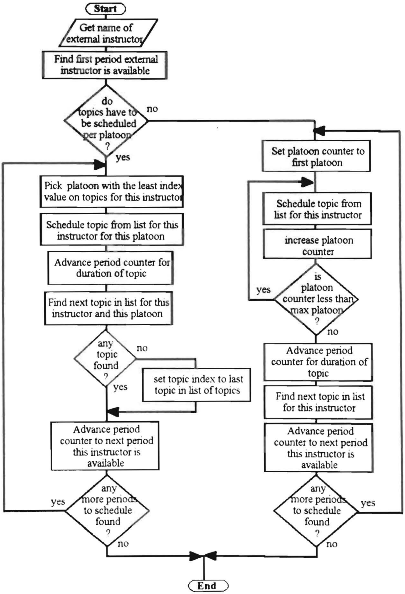

Figure 3: Flowchart for Scheduling Algorithm for External Instructors

2. Prescheduling

The two major training facilities can be prescheduled because topics requiring them can only be taught during their availability. Additionally, if this kind of facility is assigned for use, then the company has to use it. Not using this facility could lead to less time the following quarter. If the facility is not used it has to be justified to the superior command as well as to the organization responsible for assigning the facility. The scarceness of these resources justifies the approach of scheduling a topic to use the facility whenever it is available. Considering only the times of availability for the given facility the prescheduling process assigns topics in sequential order to the schedule observing given rules in the catalog of training objectives.

a. Garrison Training Area

The schedule for all platoons can be the same. Therefore, they can be assigned simultaneously. Each day of training has a given structure that uses the first period of a day for a preparation topic and the last for a clean up topic. For training that starts on one calendar day and goes without break to another calendar day, only the very first and very last period have the special topics.

The objectives from the catalog which have to be done on the Garrison Training Area can be grouped. Each group has a given earliest start time specified in weeks on the CTO flowchart. These earliest start times are converted into actual periods and then scheduled following precedence rules. The rules are also illustrated in the flowchart in Figures 4a to 4d. The precedence rules determine that anti-tank weapon topics are assigned prior to engineering topics which in turn are assigned prior to infantry topics when multiple topics are available for scheduling. In infantry training objectives the training and exercise marches have to be treated separately and take precedence over other topics when scheduling conflicts exist. If objectives cannot be scheduled entirely in a day, they are not divided but are scheduled on the next day of training and other topics are scheduled in lieu.

If training with limited visibility conditions has to be scheduled and there are not enough objectives available for scheduling because the accompanying daylight objectives have not been scheduled, previous objectives are repeated until the required daylight topics can be scheduled. If there are more periods on the Garrison Training Area in the schedule than training objectives, the last and therefore, most sophisticated topics are repeated for training and drill. On the other hand, if less time is available than needed, high end topics are left out.

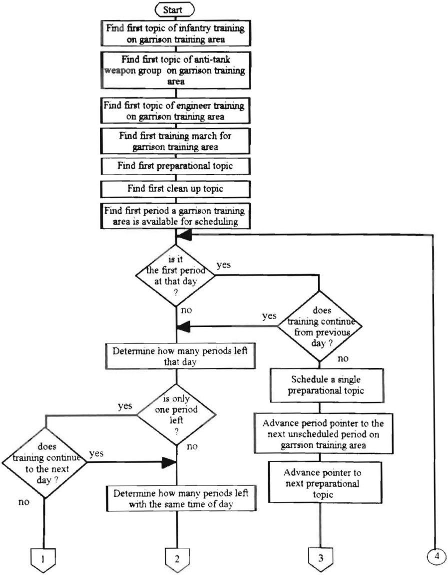

Figure 4a: Flowchart of Scheduling Algorithm For Garrison Training Area

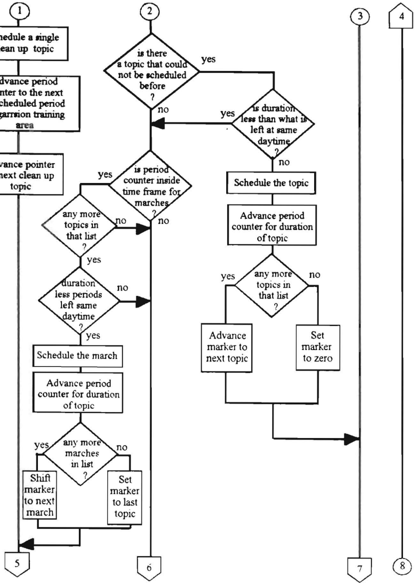

Figure 4b: Flowchart of Scheduling Algorithm For Garrison Training Area (Continued)

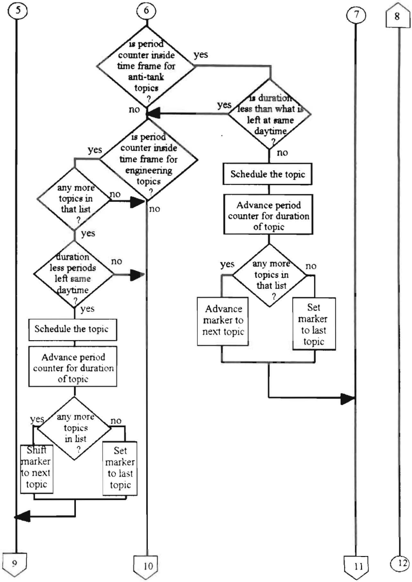

Figure 4c: Flowchart of Scheduling Algorithm For Garrison Training Area (Continued)

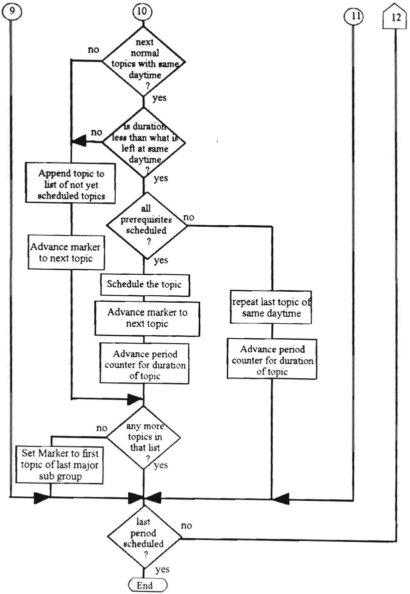

Figure 4d: Flowchart of Scheduling Algorithm For Garrison Training Area (Continued)

b. Firing Ranges

When scheduling Firing Ranges each platoon has to be treated separately and each subrange has to be considered. Scheduling Firing Ranges is similar to the Garrison Training Area scheduling previously discussed. The appropriate flowchart is shown in Figures 5a to 5d. The CTO objectives associated with Firing Ranges can be grouped, but the main gunnery objective group must be subdivided further to capture the differences in range length needed for different weaponry.

Due to the special nature of the anti-tank weapon firing exercise and the firing exercise for security guards, they have to be treated individually and are scheduled first. The program always schedules the firing exercise for the security guards on the last day the required range size is available and then schedules each platoon to take turns on that range. This is justified also from the flowchart which shows this training objective at the very end of the quarter. The anti-tank weapon firing exercise is then scheduled on the last day a night Firing Range is assigned to the company. Here the platoons are also scheduled to take turns on that range.

All other training objectives are then scheduled beginning with the first range available. Prior to any scheduling the platoon whose exercises get scheduled first has to be determined. The platoon that has the fewest objectives scheduled is scheduled for live firing. For this assignment each platoon is then assumed to stay on that range the whole day. This assumption is valid because, in general, the time needed to get to the range and get back to the barracks is significant. This process is repeated for the day as long as subranges are available. Then the next day is scheduled.

When all objectives are scheduled the program checks for blanks in the schedule on Firing Ranges and inserts into these blanks training objectives that give freedom to the platoon leader. He then can choose another live firing exercise or repeat an exercise for drill.

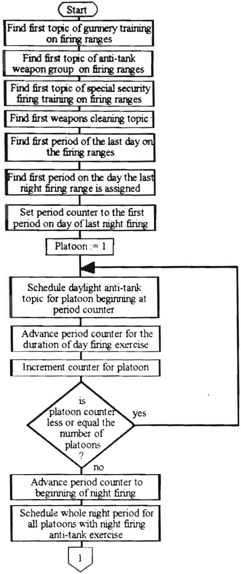

Figure 5a: Flowchart of Scheduling Algorithm For Firing Ranges

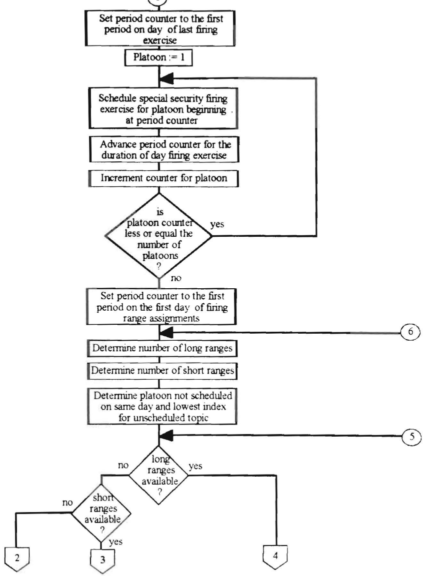

Figure 5b: Flowchart of Scheduling Algorithm For Firing Ranges (Continued)

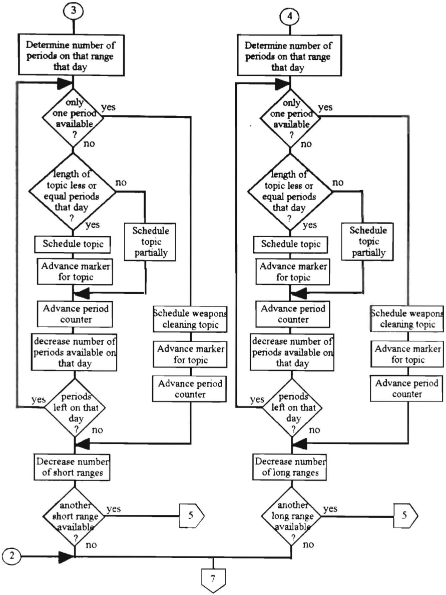

Figure 5c: Flowchart of Scheduling Algorithm For Firing Ranges (Continued)

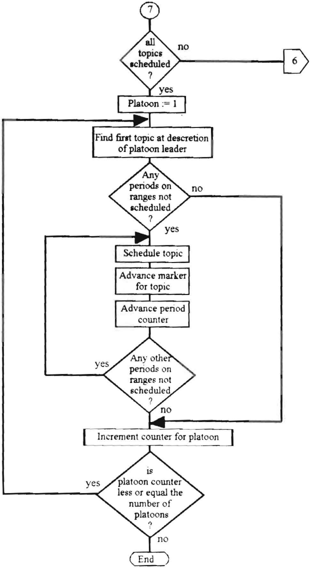

Figure 5d: Flowchart of Scheduling Algorithm For Firing Ranges (Continued)

3. Scheduling Heuristic

The heuristic involves a sequence of steps that are repeated until the last period has been scheduled. Initially, the period counter for each platoon is moved to the first unscheduled period for that platoon. Then the platoon that has the smallest number is chosen to have the next topic scheduled. The scheduling is then performed beginning with this period and then advancing the period counter and making decisions to find the locally best solution. This approach generates a mixture of topics among the platoons and prevents one platoon from having a superior schedule.

To determine which topic is the next to schedule a selection process is employed. This selection process starts with the calculation of the earliest and latest period a topic can be scheduled. These values are calculated for each unscheduled topic. To determine the earliest possible period the suggested start times from the flowchart in the CTO are considered. If a topic has no prerequisite and is not part of a chain of topics that built on each other this value is used. If the topic has a prerequisite that is already scheduled, the calculation checks whether the prerequisite ends later than the suggested start time from the flowchart. In this case the value for the earliest possible period are modified. A topic that is part of a chain of topics has the value for earliest assignment either from the flowchart or, depending on its position in the chain, the first unscheduled period following the total duration of all previous topics in the chain. If the direct predecessor in the chain has been assigned, the same rule as for prerequisites applies.

The latest possible period for scheduling can either be the value from the flowchart in the CTO or a value that depends on other topics. If a topic is a prerequisite for an already scheduled topic the value is the minimum of the flowchart value or the period number preceding the topic it is a prerequisite of such that it can be scheduled for its total duration. If part of a chain of topics, it is the minimum of the flowchart value and the period determined by the total duration of all later topics in the chain. If a topic has a latest value from the flowchart, the latest possible period to schedule is the unscheduled period prior to that value that allows scheduling of the whole duration of this topic prior to the flowchart value. In a chain, all later topics have to fulfill this requirement.

At this point in the algorithm any possible conflicts can be identified. A conflict can be detected if the earliest possible time for scheduling is later than the latest time. The algorithm should try to resolve the conflict by backtracking. The backtracking is not implemented in this version of the heuristic.

After finishing the calculation, topics for scheduling have to be selected. The algorithm has a set of selection rules to determine the possible topics for scheduling. Those are given in the following pseudocode listing. Simply choosing a topic might not be enough; depending on the rule, there can be multiple topics or none at all that meet the criteria. All topics should be assigned as early as possible to maintain a buffer for other assignments.

A topic is said to have the least freedom for scheduling if the difference between the value in the array for earliest possible scheduling and the one in the array for latest possible scheduling is minimum. The one with the highest penalty is the one with the highest fixed penalty in the coded CTO or the one with the highest dynamic penalty, where the diminishing freedom of scheduling increases the penalty. The selection rule of longest duration is self-explanatory.

If the candidate array is empty or all topics in the array belong to the group of general topics another rule is applied. The topics that have the smallest value in the array for earliest possible scheduling are considered to be eligible and they are chosen using the same selection rule. This enforces additional precedence of other topics over the topics in the group of general topics.

<u>Pseudocode for the selection process:</u>

```
Get the current clock (period) value for the platoon;
Initialize the eligible set and the counter for topics in set to 1;
while not all topics have been tested do  {determine the first entry}
   if current clock value = entry in earliest array then
      case selection rule of
         least freedom : if (least freedom of topic > least freedom of topic in
                                          eligible set) or (eligible set empty) then
                              enter topic into eligible set or replace old one;
         highest penalty: if (penalty of topic > penalty of topic in eligible set) or
                                    (eligible set empty) then
                              enter topic into eligible set or replace old one;
         longest duration: if (duration of topic > duration of topic in eligible set) or
                                    (eligible set empty) then
                              enter topic into eligible set or replace old one;
      {contains now one single entry, now find possible ties}
for Index = Index of topic in set to last topic do
     if (current clock value = entry in earliest array) and (value of selection rule
              matches value of topic in eligible set) then
        increase counter for topics in set;
        append topic to the set;  {contains now possible ties}
```

if (Counter for topics in set = 1) **or** (the first entry = 0) **or** (first topic belongs to main group of general topics) **do**
find another topic that is closest to the current clock value using the same selection rule;
{now contains topics}

If more than one topic is listed in the set of eligible topics those are called ties. A decision has to be made which one to choose. As shown in the pseudocode there are multiple tie breaking rules available in the program. The other way of breaking ties depends on the selection rule for the topics. The selection rule determines the sequence of tie breaking rules that are applied until a unique selection for scheduling is be made.

```
Pseudocode for the tie breaking process:
while there is no unique topic to schedule do
   if tie breaking rule is random do
      pick a random number from the range of indices of the eligible set;
      chose the topic at that index;
   else
      case selection rule of
        least freedom : choose the topic with the highest penalty;
                        if not unique then
                           choose the one with the longest duration;
                        if still not unique then
                           choose the first of the remaining;
        highest penalty: choose the topic with the least freedom;
                        if not unique then
                           choose the one with the longest duration;
                        if still not unique then
                           choose the first of the remaining;
        longest duration : choose the topic with the least freedom;
                        if not unique then
                           choose the one with the highest penalty;
                        if still not unique then
                           choose the first of the remaining;
```

After determining the unique topic to be scheduled, it is scheduled for its whole duration. Then it is flagged as scheduled and the clock for this platoon advanced to the next unscheduled period. Now the next platoon that has the least value for the clock is chosen and the process reinitiated until all periods have been scheduled. The algorithm then calculates the measures of effectiveness for this schedule and writes the report that includes the calculation time for the algorithm.

The choice of a random tie breaking rule has to be investigated closely. Without changing the data the solution from one calculation to another would change. The evaluation of whether the result is better than results using deterministic rules has to be made after test runs. Chapter V.A shows the differences between those rules.

4. Improvement Algorithm

The improvement algorithm employed here is an interchange algorithm that tries to interchange topics or blocks of topics of equal length in order to improve on the measures of effectiveness. The outline of the algorithm is shown in the flowchart in Figure 6.

Before making any changes to the given schedule the algorithm calculates the current values for all MOE's and saves them. Then the algorithm considers each platoon individually for all those topics that are not bound to a facility or an external instructor. It chooses the first topic and then tests whether this can be exchanged with the next topic or topics that match the total duration. Topics can be exchanged if they are not prerequisites of each other and none of the topics between them is a prerequisite of the originally later one.

If they can be exchanged, the program exchanges them and recalculates the measures of effectiveness. If the measures of effectiveness have improved, the exchange remains otherwise the exchange is not accepted.

In both cases now the period counter is advanced to the next period following the end of the first topic chosen for exchange. Now this process continues until all topics for all platoons have been tested. The improvement algorithm tests only locally possible changes. Therefore, after an exchange additional opportunities might arise for exchanges of earlier scheduled topics. To investigate this possibility the algorithm is employed multiple times to determine those effects. The algorithm can be repeated until no improvements in the measures of effectiveness occur.

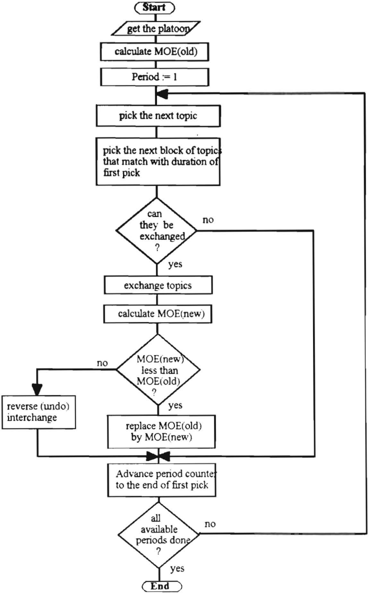

Figure 6: Flowchart of Improvement Algorithm

C. TEST PLAN AND TEST DATA

To determine the best combination of selection rules and tie breaking rules to pick the next topic for scheduling a number of different combinations were tested. To make the combinations comparable they were done on the same data set. This data set is listed in Appendix G. It also illustrates the data needed from the user. The test plan checks all implemented combinations. The tested combinations are given in the first seven columns of Table 4 and 5 in Chapter V.

In addition to these tests the number of replications for the improvement algorithm has to be determined. The algorithm is employed once, twice, three times or more until the improvements from one iteration to another reach a threshold or until the computing time exceed the actual runtime of the clock advancing algorithm.

After determining the best combination of selection and tie breaking rules in combination with the way of calculation any penalties, the stability and performance of the program for different data sets was tested. Tests were done on different quarters of the year to evaluate the impact of the length of the quarter and of moving legal or clerical holidays. To compare different quarters, the number of days the key facilities are available is held constant. External instructors are available for the same number of periods. It is not possible to insure that the assignment of facilities always fall on the same training day in each quarter. Depending on days without duty they have to be shifted. The objective in this test is to compare calculation times for the quarters and the performance in terms of scheduling. The major interest is in the topics that could not be scheduled and those that are scheduled outside the suggested time frame.

IV. ENVIRONMENT AND USER INTERACTION

This section describes the program and the design decisions that governed the development. The pseudocode is included only if vital for understanding. One basic underlying design decision is the use of a DOS based personal computer. This system is on the computers in the field and a graphical operating system cannot be expected in the near future. The TURBO VISION (Borland, TV, 1992 and Watson, 1994) engine which is included in TURBO PASCAL provides basic functionality for building a graphical user interface (GUI) as the front-end on which any customization can be build. The basic functions generate a GUI that is similar in look to any DOS based program GUI the user might have seen earlier.

A. USER INTERFACE

A GUI is easy to use if it includes intuitive menus and sub-menus. The learning curve of familiarizing oneself with a GUI is normally not steep. Another big advantage is that there is no need for memorizing commands and short-cut keys before the program is usable. To avoid any difficulties with typographical errors in script files or data files, all data are entered using dialog boxes. They provide clearly labeled entry fields and confirmation or choice buttons. Erroneous input can be detected at this stage. The clear labeling of choices and input fields reduces the learning curve. The program displays context-sensitive help either in the dialog box or on the status line which is the last line of the display.

During design of the GUI emphasis was placed on consistency. The main design has to follow standard practice of commercial programs. Naming conventions for menus and display objects on the screen should be familiar to the user. Dialog boxes that call for similar information are designed similarly. Buttons with the same label perform the same action. Placement of the buttons is, whenever possible, at the same position in each dialog box. Message boxes with warnings or error messages have only a confirmation button. Choices or actions that are currently not available are disabled and are automatically enabled as soon as they become appropriate. The display of context sensitive help in the status line of the display supports ease of use.

The program has dual operationality with the keyboard and with a pointing device. This eliminates the sole reliance on the use of a pointing device, mouse or trackball. The system has redundancy built in. All actions on the screen such as activating menus, calling up of dialog boxes, moving from one entry field to another and activating buttons are possible with the keyboard. Using the keyboard for moving on the screen is consistent with commercial applications. For moving from one record to the next in a dialog box, commonly known keys PG UP and PG DN are used. The functions of buttons are described here only if their function is more complex than the label indicates.

The main screen provides the menu and the status line as well as the common background for all dialog boxes. Figure 7 illustrates the appearance of the main screen with no menu activated and with no dialog box. The following menus are available:

- File
- Edit
- Run
- Change
- Show / Print
- Preferences
- Window
- Help

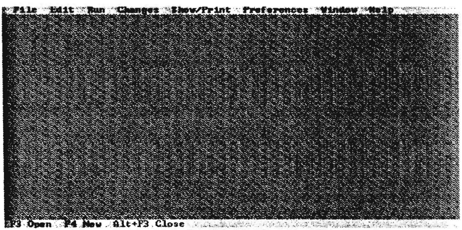

Figure 7: The Main Screen of the Graphical User Interface

The "File" menu provides general file handling procedures and the termination command for the program. Under the "Edit" menu the user finds the basic commands to

do file editing and special commands to call up the dialog boxes needed to supply the data. These are explained in detail in section B. The "Run" menu contains the commands to start execution for calculating the schedule and a calendar. The "Change" menu which is only accessible when a schedule is displayed provides the commands for user interaction and modification. For discussion of these features see section D. "Show / Print" contains all commands for displaying results on the screen and printing paper copies of all the results see Section C. The "Preference" menu gives the user limited influence on the appearance of the main screen and lets the user store and retrieve his settings. "Window" makes basic commands available for moving and sizing windows on the screen. "Help" only contains an "About" command that displays a message box about the purpose of the program. An On-line-help is not implemented. All menus are displayed in Appendix D.

B. DATA ENTRY AND VALIDATION

The data input for the scheduling program is basically manual via dialog boxes which prompt the user for the necessary information. The catalog of training objectives is provided by the program and the user is not required nor should he try to modify or augment this catalog. Any changes to the catalog such as omitting training objectives have to be done with manual input into dialog boxes. Major changes in the catalog need to be made centrally and then distributed as a file to the individual users. The program needs the following information from the user:

- basic weekly schedule of the company,
- main calendar dates, such as quarter start, quarter end, holiday and duty on Saturdays or Sundays,
- special events not captured in the catalog of training objectives,
- availability of training facilities,
- availability of outside instructor personnel,
- non-availability of company instructor personnel,
- additional requirements from division or battalion level referring to the topics listed in the catalog of training objectives.

The program can manage two quarters simultaneously: the current quarter and the next quarter. The current quarter is determined by a check against the computer date. The program keeps a log-file to determine last use of the program. A quarter change between two consecutive uses of the program results in a renaming of files belonging to

the former current quarter into files belonging to the old quarter. These files are stored for backup reasons and are not accessed by the program. The files belonging to the former next quarter are then renamed as files belonging to the current quarter and new files will be made for the then new quarter. These changes are made automatically and the user has no need to concern himself with them.

The following sections describe the format for the dialog boxes and the information that is entered into them. All these features are menu items in the "Edit" menu. Some of the menus have sub-menus associated with them if further differentiation is necessary.

1. Basic Schedule

With this dialog box the user has to supply the basic schedule for the company. The required format for each period on a day with duty, usually Monday to Friday, is start time of the period - end time of the period in 24 hour military notation, utilizing the following picture ####-####. Any other input is rejected prior to saving. The input of characters is disabled. Single digit hour values need the leading 0 in their input. For each day from Monday to Thursday the screen provides a maximum of ten slots to fill and for Friday a maximum of six slots. After initial input of these data each time the user calls this menu the previous input is displayed. Figure 8 shows a filled basic schedule matrix with a typical company schedule.

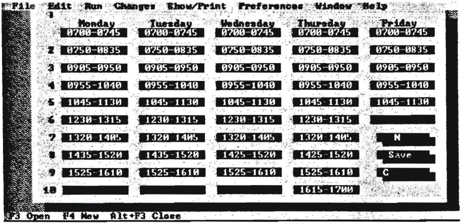

Figure 8: Basic Schedule on Screen With Sample Data

The three buttons in the right column perform the following actions. The "NEW" button clears all fields and allows total new entry. The "SAVE" button saves the current displayed data after format checking and the "CANCEL" button closes the dialog box without saving. This window as any other can be closed also using the small square button at the upper left or the keyboard shortcut combination "Alt" + "F3". The plus sign symbolizes the simultaneous engagement of those two keys.

2. Instructor Availability

This menu provides access to the dialog boxes for all external and company instructors. Therefore, a sub-menu handles the choice of the instructor for whom the user wants to enter the data. After choosing the instructor, the user is led to a detailed input screen where he can provide the actual information about the availability of the instructor. In the case of company personnel as instructors the user has to stipulate when the instructor is not available. One instructor dialog box is displayed in Figure 9, the others, as well as the preceding choice menu, can be examined in Appendix D. Each dialog box contains a line of text to remind the user which information is needed. Each dialog box looks similar and requires the input of a date and time.

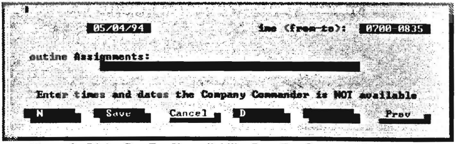

Figure 9: Dialog Box For Unavailability Data For Company Commander

The dialog boxes for the company instructor personnel also have check boxes to ease the input of recurring data for daily, weekly or monthly events. Only the information about the first of those events need be supplied by the user; after checking the appropriate check box the computer calculates all other data and inserts them according to their time sequence.

For this calculation the computer stores the dates temporarily ignoring possible days without duty. Although those dates are kept, any conflicts are resolved later during

coding of this information into the period equivalent. This approach is used to maintain the freedom for the user to supply the data in whatever order he likes. Other approaches would have required that the input of the calendar dates be made prior to entering any other information. The program in this stage of input does not check for overlaps in the given times. Any overlaps are sorted out and eliminated during the coding phase.

Only button options that might perform meaningful actions at a given time are available. If the user wants to delete a record he has to move through the list to display the record on the screen. Only then he can delete this special piece of information. Data created by the program using the check boxes cannot be deleted as a whole. Only the current visible data record is deleted. This design is based on the policy that nothing the user cannot see should be deleted.

3. Training Facility

After engaging this menu item the computer displays a choice menu for the different facilities. After choosing the specific facility the user supplies the date and time of availability. The basic operational principles in these dialog boxes are the same as for the instructor availability dialog boxes. Only two of the boxes are equipped with additional check boxes for repetitive data: the gymnasium / athletic field and the obstacle course dialog boxes. These facilities are likely to be assigned to the company on a regular weekly basis. To avoid many different styles of dialog boxes, the check box area, if required, is the same for all instructor and facility data entries if required for the instructor or facility.

The only dialog box that has a unique feature is the one for Firing Ranges as shown in Figure 10 (the other dialog boxes are displayed in Appendix E). The user has to supply the subranges that are assigned for a given day. These entry fields require an integer number. The program automatically checks the range of those entries and rejects numbers greater than 6 for each range. Higher values are most likely an input error because Firing Ranges that belong to the garrison are usually small and have at most six ranges of a given subtype. Some garrisons might have fewer subranges. The program does not include personalization such that the actual local conditions can be entered to determine the validity check for the range numbers. The restriction to a range from 1 to 6 maintains usability for all potential users.

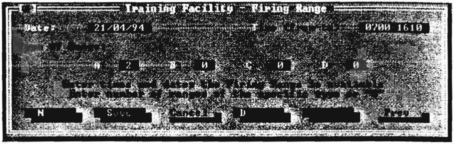

Figure 10: Input Dialog Box for Firing Range Data

The present catalog only makes use of subranges of type "A" and "D" but to allow for changes in the catalog of training objectives or the accompanying live firing exercises the subranges "B" and "C" are included in the interface but are not implemented. If the CTO changes, the user need not learn a new interface.

For deleting an entry the same rules apply as described above for the instructor availability.

4. Battalion/Division Requirement

Additional guidance or demands for the schedule given to the company from higher commands can be entered using this dialog box. This dialog box as shown in Figure 11 should be used only to give guidance about topics in the catalog of training objectives. Therefore, other input in the first column of input lines is not accepted. For each topic the user can enter the days on which they should be scheduled or should not be scheduled. The day has to be given with its name, i.e. Monday, Tuesday etc. Checking the a.m. or p.m. check box then limits the possibilities for scheduling to the appropriate part of the day. Both checked or both unchecked is treated as if the whole day is available.

Additionally a topic can be scheduled manually by giving the exact date and time for scheduling. The date and time format is consistent with any other date and time format used in this program. As the last option the user can single out topics which should be in a week with at least n repetitions or should be in a week with no more than n repetitions. This field accepts only small positive integers.

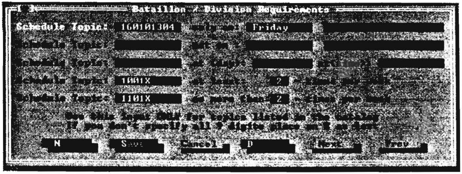

Figure 11: Dialog Box For Battalion / Division Requirements

5. Other Input Menus

In addition to the menus described above the program utilizes the following input dialogs:

- quarter choice,
- main quarterly calendar data,
- special events,
- number of platoons and anti-tank weapons present in company.

The quarter choice is automatically presented when the program is started. During run-time it is possible to change the working quarter with a menu-item under the "Edit" menu of the main screen. As displayed in Appendix D the last menu-item again brings back the initial choice box as shown in Figure 12.

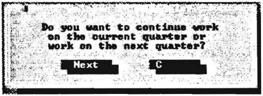

Figure 12: Initial Choice Box to Determine Quarter

The main quarterly calendar data consist of the arrival date of the new recruits, the departure date, the dates of Saturdays and Sundays that are planned for duty and the dates of clerical and legal holidays. The last input section also provides the opportunity to supply to the program pre-planned days off, leave for special occasions or leave as

compensation for additional duty. All these dates have to be entered in the format Day/Month/Year, where only the last two digits of the year are entered. For the day and the month always two digits have to be given. The leading 0 is mandatory. There are possibilities to enter up to 15 dates for holidays and up to five dates for Saturday / Sunday duty. This dialog box is displayed in Appendix E.

Occasionally there is the need for events or training objectives that are not covered in the catalog of training objectives. These can be entered into the data via the appropriate dialog box as displayed in Figure 13. The title has to be provided to the program for later printing of a schedule that contains titles and instructors for topics instead of nine digit numbers. For date and time the common format restrictions apply. The instructor has to be given in the military abbreviated form as used on schedules, not in numerical coding as used internally by the program.

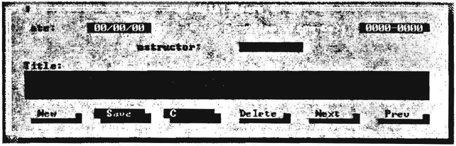

Figure 13: Dialog Box For Special Events or Special Hours

At the start of execution of the scheduling algorithm the program prompts the user for confirmation or change of basic information about the company. These data are considered shared data over quarters. The displayed dialog box as shown in Figure 14 lists the number of platoons in the company. This input line accepts only 1, 2 or 3. The check boxes in this dialog box are for the selection of the anti-tank weapons the company uses. A checked box indicates the use of that weapon. The user can check none, one, two or all three item in the selection. Only the topics related to the chosen weapon is scheduled. If no box is checked none of the topics related to anti-tank weapons is scheduled.

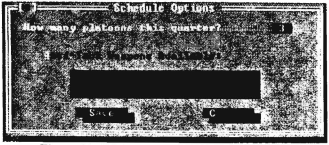

Figure 14: Dialog Box At Start of Calculation

6. Data Checking

Prior to accepting data, validity checks are performed. The date and time formats are checked and if they do not conform to the given, natural formats they are rejected. The user uses the familiar format for date : ##/##/## and for times: ####-####, where the "#" stands for a single digit. There is no check whether the date or time given to the computer is a valid one. The program accepts a time 9992-0193 for example but during the calculation in subsequent steps simply disregarding this invalid input.

The data for the basic schedule of the company are not likely to change from quarter to quarter as other data do. This means the basic schedule information needs to be input only the first time the program is used or if data change. The basic schedule is considered shared information between quarters and is automatically used for the calculation.

For training facilities and platoons, range checks of the supplied values are used to determine any unreasonable large numbers. For example, for platoons the numbers 1 to 3 are the only accepted input.

The user has complete control over the order in which he enters the data. There is no given sequence he has to follow; he can open any dialog box if he missed a step or wants to change a previous input. Data that have a natural time sequence order is automatically ordered by the program. This allows for further flexibility to enter the data whenever they become available. Augmenting or changing data is done with the same input features as before, information already entered is displayed when this data entry screen is opened.

Because no meaningful schedule can be produced without the data about training facilities, the program checks for their presence. If they are not present or cannot be

accessed, the algorithm alerts the user with an error message before doing any calculation and aborts the process to allow the user to provide the necessary information. The error and abort messages are displayed in Appendix F. The program does not provide a manual override at this stage because any calculation would be meaningless.

C. PROGRAM OUTPUT

The program provides the user with three different kinds of output. All three kinds of output can be shown on the screen and are printable in the same format on an attached printer. On-screen displays for platoons that are identified by shaded buttons are replaced in the printed version by a text line that identifies the platoon. The three outputs are:

- quarterly calendar with main dates and facility assignments (three screens or pages);
- weekly schedule, divided by platoons (number of platoons times number of weeks in the quarter, screens or pages), that is, three platoons and 13 weeks result in 39 pages;
- report on scheduling progress, containing unscheduled topics and the measure of effectiveness data.

This output cannot be edited or manipulated by the user directly. Any changes to the schedule have to be done using the menu for user interaction, described in section D. The calendar data and the result are prepared by the program and can be printed. Figure 15 shows the on-screen display of a monthly calendar and the print version is shown in Appendix A. The calendar data are stored in an ASCII file and could theoretically be altered using the file editor but the program would not take any of those changes into account. Any new call to the procedure for calculating the calendar would restore the previous result.

The schedule data consist of the numeric code representation of each topic and can be displayed one week at a time. Figure 16 shows the appearance on the screen and Appendix B has two weeks in their printed version.

The report data are written during runtime of the calculation and are valid only for the current calculation. Any changes in the base data or conditions for scheduling would produce a different report. The report provides the user with indications about problems

during scheduling and the methods the program used to solve conflicts or violations of constraints. A complete report is listed in Appendix C.

The printer output is initiated by either dialog boxes (Appendix E) or directly from the display on the screen (Figures 15 and 16). The user has to supply the number of copies and the output range. For the calendar output he can specify a month or the whole quarter. For the schedule a single week for a platoon can be printed or the whole schedule for a single platoon. Another option is to print a single week for all platoons. The last option is to print the whole schedule for all platoons. When printing from the display on the screen the user can print only a single copy. If the calendar is displayed, the button "PRINT" initiates the printing for the month currently on display. The button "PRINT ALL" initiates the printing of all three months of the quarter. If a schedule is displayed the user can print the displayed week of the platoon shown by using the "PRINT" button. The button "PRINT ALL" produces a printout of all weeks for all platoons. The report cannot be printed from the display.

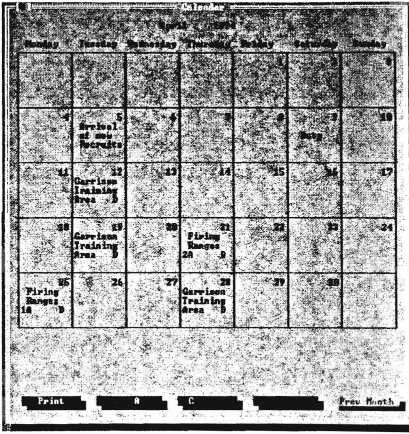

Figure 15: Screen Display of the Calendar

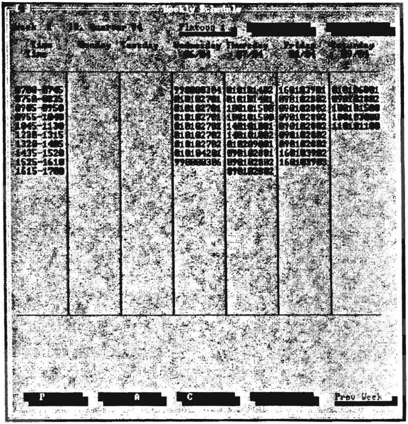

Figure 16: Screen Display of A Schedule (Platoon 1, Week 1)

D. USER INTERACTION

After the computer construction of the schedule the user can manipulate the schedule. The following options are available only when he displays the schedule on the screen. The sub-menu is displayed in Appendix D. It consists of the following choices:

- interchange hours;
- interchange days;
- facility unavailability;
- instructor unavailability;
- add / remove special hours.

Any changes the user tries to make have to be done using the special change and modification dialog boxes. The dialog boxes used prior to the construction of the schedule have no effect. Any changes in those data are not used during manual changes. They can only be used to enter data and then recalculate the schedule. The user must realize that any manual changes are lost if the schedule is recalculated. Therefore, after having made manual changes, any further changes should be made manually. During the manual changes the program uses only the data the calculation of the schedule was based on.

Each of these choices brings up a dialog box for input. The dialog boxes are displayed in Appendix E. The requested action is tested to determine whether it is possible to perform it. The checks include checking the prerequisites. The algorithm checks whether the exchange would lead to a schedule where a topic is scheduled earlier than any prerequisite. If this is the case a message is displayed and additional confirmation requested. The user can override the program and force the change. He then takes the responsibility to introduce a chain of changes to fix this violation of constraints. The override is necessary because the program cannot determine in advance all steps the user wants to take and perhaps allowing only legal action would make it impossible for the user to accomplish desired changes. Most of the changes only influence the schedule for the platoon that is currently displayed on screen. The exception is facility unavailability if it concerns Garrison Training Areas. Because they were scheduled simultaneously for all platoons an unavailability has to be dealt with for all platoons simultaneously.

The interchange of hours is initiated by entering the information in the dialog box. The program then tests whether the requested change is for periods of equal length in

both cases. If not, the algorithm stops and displays a warning box to the user that provides no override possibility (Appendix F). If they have same length, the program checks whether the earlier is a prerequisite of the later one. After passing this test, the next check involves testing whether one of the topics between the earlier and the later is a prerequisite of the later or has the earlier as a prerequisite. In those cases the program would display another message box and the user must confirm this interchange and force the program by override to perform. If none of the tests indicate a problem or the user overrides, the program performs the interchange. The pseudocode of this algorithm is listed in Appendix H.

The interchange of two days involves the same checks for violations as the interchange of hours. If the first check returns that the days have an unequal number of periods, the warning box allows a manual override from the user (Appendix F). Forcing the manual override initiates an interchange of the periods on both days that match. The program adds to the day with the fewer periods as many additional periods as needed to accommodate the topics from the longer day. The time frames for these periods match those from the longer day. Then the topics from the longer day are reassigned to the new periods and for this platoon those periods on the longer day are coded as not scheduled. The display shows the number "0". For the platoons not involved in the interchange the topics on those new periods are coded as not scheduled. For the pseudocode see Appendix H.

If a key facility is no longer available to the company in the quarter, the topics need to be rescheduled or dropped. The two major facilities, Garrison Training Area and Firing Range, have to be treated separately. The other facilities require less work in the algorithm. For all facilities the user supplies the date and time it is no longer available and checks in the dialog box the name of the facility (Appendix E). If he marks more than one facility the program stops and displays a warning that only one facility is to be marked at a time (Appendix F). After the user confirms this warning he is shown the interchange data and can correct his input and restart the algorithm.

For Gymnasium, Swimming Pool and Obstacle Course the algorithm unschedules the topics at the given periods of the schedule. Then the list of topics for this platoon is scanned for other unscheduled topics. If a topic is found that requires another facility which is available and unused that topic is scheduled. If not, the program tries to schedule another topic from the list of topics out of other groups, then from the main group of general topics. If the program could not find an unscheduled topic or could find

only topics with duration greater than the number of periods to be replaced, the program repeats topics from the main group of general topics.

Garrison Training Areas and Firing Ranges have to be handled differently. Scheduled topics on those are closely linked and are usually part of chains of prerequisites. Therefore, if the unavailable day is not the last day, prerequisites must be checked. After the data have been supplied, the program determines whether it is the last day. If it is the last day, the topics have to be dropped and are replaced by unscheduled topics. The way of replacing them is the same as for the other facilities. If the day is not the last, a replacement and shifting of the topics has to be done. Both parts are shown in the flowchart in Figure 17 and Figure 18. Due to the complexity of decisions on the replacement on the actual day the facility is no longer available these processes have been separated and they are shown in Figure 18. The topics on the day in question are temporarily stored if they do not belong to the topic group of general topics. Topics from the group of general topics are only unscheduled. Prior to saving the other topics on the facility the algorithm tests whether they are scheduled also on another day. If they are repeatedly scheduled, they are not shifted. The actual day the facility is no longer available is treated the same way as the last day. If the list of saved topics is not empty they have to be shifted to the next day this particular facility is available. The program determines the total length of those topics and tries to put them in as early as possible. The topics on the next day that are displaced by those earlier topics are tested the same way and saved for shifting if they are uniquely scheduled. Then the earlier topics are scheduled as replacement. This procedure continues until either the list of topics to shift becomes empty or the procedure reaches the last day and has topics left on the list. If topics are left on the list after the last day, they are marked in the data as unscheduled.

Dropping the assignment of an instructor to a given period because of his unavailability involves fewer steps. The topics from external instructors are those the user has to be concerned with here. Company instructors usually only require interchanging of periods. The topics to be given by external instructors have no prerequisites and also do not built on each other. Therefore, the exchange simply involves unscheduling that topic and replacing it by another topic the same way a replacement is done for the topics on the lesser facilities. The pseudocode is shown in Appendix H.

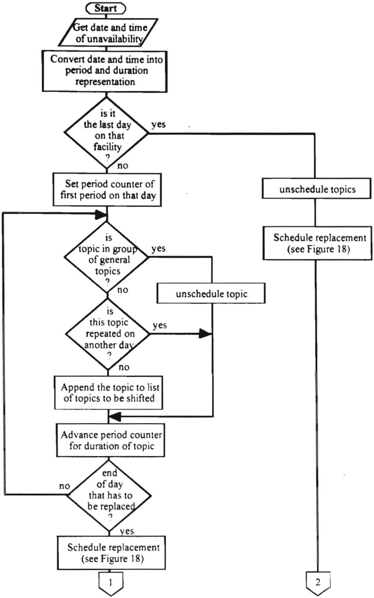

Figure 17a: Flowchart of Algorithm To Handle Facility Unavailability (Garrison Training Area or Firing Range)

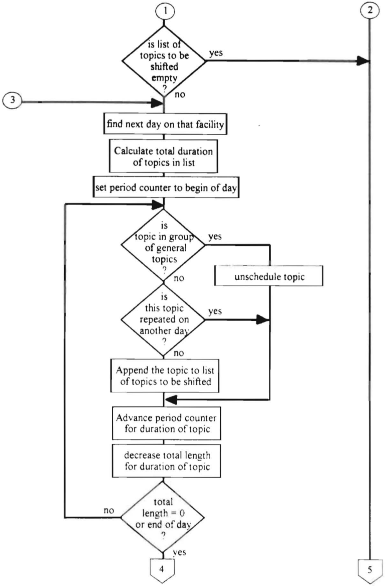

Figure 17b: Flowchart of Algorithm To Handle Facility Unavailability (Garrison Training Area or Firing Range) (Continued)

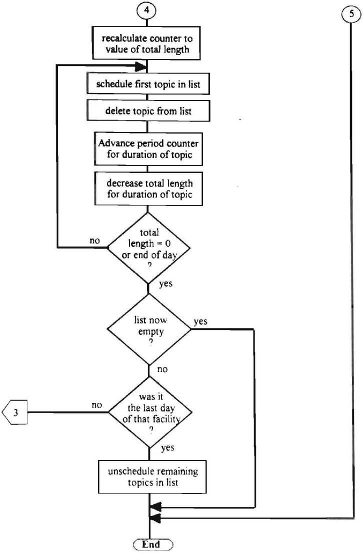

Figure 17c: Flowchart of Algorithm To Handle Facility Unavailability (Garrison Training Area or Firing Range) (Continued)

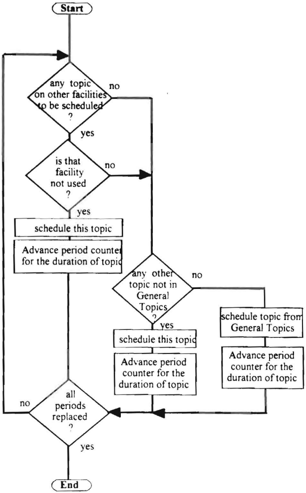

Figure 18: Flowchart of the Decision Processes in the Replacement Scheduling

A topic not included in the CTO can be added to or removed from the schedule using the last menu option. If a special event is to be added, the algorithm first determines how many of these topics are already in the schedule and names this new topic "Special" and augments the name by a number that is one higher than the number of those already included. The topic including the information about the period it must be scheduled is then appended to the list of topics and the algorithm increases the total number of topics considered. The program scans the existing schedule and locates the topic originally scheduled at that period. If this topic is not part of a chain for prerequisites it is unscheduled and the new special topic is scheduled in its place. If it is part of a chain of prerequisites, the program tries to find a period prior to the topic it is prerequisite to and tries to schedule it in place of another one with no prerequisites or a topic from the last main group, the general topics. If it is only possible to reschedule this topic at the period the topic it is prerequisite of is scheduled, the program performs this scheduling. Then the process is repeated with the topic that had to be removed.

Removing a special event or hour simply unschedules the one in question and replaces it by employing the same decision rules as used for a replacement if a lesser facility becomes unavailable. The whole algorithm is listed in Appendix H as pseudocode.

After each exchange operation the schedule is updated and the revised schedule displayed. The new schedule shows then the same week and same platoon as previously displayed.

V. RESULTS

The results are shown in three steps. First, the computational results are evaluated together with the determination of what is the best combination of selection rules. Next, a possible scenario of use illustrates the steps the user has to take and estimates the amount of time needed. Finally, the benefits of having this program in use with reflections about the acceptability of a computer generated schedule are given.

A. SCHEDULE CALCULATION

The program was tested to determine the best combinations of selection rules and tie breaking rules. Another consideration in the testing was to compare the effect of fixed (penalty from the CTO for not scheduling the topic) or variable (penalty from the CTO modified by the remaining freedom for schduling) penalty calculation. The MOE's were used as criteria to determine the best solution. Additionally, a comparison of the computation times was made to estimate guidelines for the user. The calculations were done on an IBM compatible personal computer with central processor unit Intel 80486DX-2 and a processor speed of 66 Megahertz. This machine was equipped with random access memory of eight Megabytes. It would have been desirable to run the calculations on a machine identical to those it is intended to run on later, but none of these were available. Therefore, the calculation times cannot be taken as absolute guidelines but as indicator about stability of performance. They will be larger on the machines the program will run on.

All the tests for the best heuristic were done on the same data set. This data set is included as a sample data set in Appendix G. Table 4 shows the results for the twelve combinations of selection rules and tie breaking rules. In addition to the MOE's, the number of topics contributing to them are counted and stated in parenthesis below the corresponding value. The shortest and longest calculation times have a difference of only 35 seconds, thus the difference is not a decisive factor. An evaluation of all possible combinations shows that the effect of fixed or variable penalty calculation cannot be determined alone. Depending on the combination of selection and tie breaking rules the effect of the penalty calculation can be reversed. Using the tie breaking rule of dependence on the selection rule, the fixed calculation performs better. With random tie breaking the variable calculation seems to produce better results. For this testing the random tie

breaking run in each combination was performed once. More extensive testing is necessary to determine whether the results are statistically significant. Although the performance of the calculation with the selection rule of the least freedom first and random tie breaking is best, it is not chosen to be the implemented solution. The stable calculation, not based on randomness, of the same selection rule but paired with the dependence tie breaking rule and a fixed penalty calculation is used. This approach is only slightly worse in the MOE's and two seconds faster in the calculation.

In addition to the final results, intermediate comparisons are made between the combination of rules. In Table 5 the effect of scheduling with and without the improvement algorithm is illustrated. The improvement algorithm has no effect on the MOE1 which is the penalty for topics that could not be scheduled. But in all combinations the other MOE's improved in general with the algorithm activated. The last measure of effectiveness which is the number of more than three consecutive lectures in a row on one day is not enforced in the improvement algorithm. Therefore, sometimes this number increases. But generally the improvement algorithm is worth employing. There is no particular combination of rules that is either especially prone to improvement by the exchange or resistant to improvement.

The testing on how many replications of the exchange algorithm should be run for each platoon was also performed on the sample data set. For the first three replications the MOE's got better but no change could be detected after the third replication. For all subsequent testing, the number of replications was fixed to be three.

After deciding on the permanent combination to be implemented, the complete algorithm was tested on different quarters. The data sets for the quarters were constructed to be as similar as possible. The following basis was chosen:

- assign 13 days of Garrison Training Areas with 3 night training parts;
- assign 6 days of Firing Ranges with 2 night firing exercise possibilities;
- have the Obstacle Course once a week;
- assign 5 different date / times for the Gymnasium as weekly data;
- have the Swimming Pool 9-times in the quarter;
- have company personnel not available 3- to 5-times, the CpCdr on a weekly basis;
- assign MedPers 6-times, SecPers 5-times per quarter;
- assign MilPriest 3-times and SocWork twice per quarter;
- include 3 special events;
- have the same Battalion / Division requirements.

With this basis the runs were performed and they are compared in Table 6. As the table illustrates the calculation time per quarter is fairly stable although the fourth quarter is shorter than the others and has 44 fewer periods than the longest quarter. In comparison, the MOE's vary a lot. The longest quarter uses almost all of the available topics but does not deliver a significant better result in the other MOE's. On the other hand the first quarter shows the worst result in the measure of effectiveness for the topics that could not be scheduled. Although the same number of days on a Garrison Training Area are assigned to that quarter, the placement of the night training days had significant influence. Because night training can only be scheduled if all prerequisites are met, early in the quarter repetitions had to be scheduled. Therefore, later in the quarter other important topics could not be scheduled. This shows that the results are sensitive to the placement of the night training assignments. If night training is assigned in weeks 4 and 8 and one additional night later in the quarter, better results could be obtained as the other quarters illustrate.

Comparing the results for a single quarter among platoons as shown in Table 7 indicate similar performance for all platoons. The results do not single out a platoon that is always the best or the worst in all MOE's. But it also shows that the schedules are in fact not equal. They have differences in all assignments except the Garrison Training Area (which is a result of the procedure of assigning topics there). It also shows that the number of topics outside a suggested time frame vary distinctively. The company controlled topics have a spread of 13 over a range from 38 to 51; this is a great difference. Other topics are more evenly scheduled. Although the number of topics that could not be scheduled does not vary significantly, the penalty incurred for those topics has a range from 517 to 813. This is caused by some topics that external instructors have to give and that could not be scheduled for one platoon as well as some sports topics.

As stated above, the calculation times should not be viewed as some absolute guidance for the times on the machine the algorithm is supposed to run on but as a measure of stability. Times will increase on slower, less powerful machines but the user will have his own time frame after performing the first calculation. Because of the stability, a large change in calculation time for any execution of the algorithm suggests an error in the input data.

The program produces the schedule with its accompanying report. They are illustrated in Chapter IV.C.7 and Appendices B and C.

Selection Rule for Topic			Tie Breaking Rule		Penalty Calculation		Penalty for Topics not Scheduled	Penalty for Scheduling Outside Suggested Timeframe	Number of More Than Three Consec. Lectures	Total Runtime	Algorithm Runtime
Least Freedom	Highest Penalty	Longest Duration	Depends on Selection Rule	Random	Fixed	Variable	[MOE1]	[MOE2]	[MOE3]	[min:sec. sec/100]	[min:sec. sec/100]
X			X		X		1946	18731	50	7:17.30	5:51.20
	X		X		X		1946	22649	73	7:37.23	5:53.01
		X	X		X		1946	23036	88	7:23.96	5:52.29
X			X			X	1946	18731	50	7:18.85	5:51.25
	X		X			X	1946	20393	64	7:24.03	5:52.46
		X	X			X	1946	22091	75	7:11.94	5:51.89
X				X	X		1946	19092	51	7:19.43	5:51.69
	X			X	X		1946	23574	73	7:25.12	5:53.93
		X		X	X		2052	29490	43	7:05.02	5:36.47
X				X		X	1946	18689	46	7:19.52	5:51.18
	X			X		X	1946	20850	67	7:21.71	5:52.30
		X		X		X	2071	31074	38	7:05.67	5:37.90

Table 4: Comparison among Different Combinations of Selection and Tie Breaking Rules and Their Influence on Measures of Effectiveness

Selection Rule for Topic			Tie Breaking Rule		Penalty Calculation		Penalty for Topics not Scheduled	Penalty for Scheduling Outside Suggested Timeframe		Number of More Than Three Consecutive Lectures	
Least Freedom	Highest Penalty	Longest Duration	Depends on Selection Rule	Random	Fixed	Variable	[MOE1]	[MOE2]		[MOE3]	
								before	after	before	after
X			X		X		1946	19343	18731	56	50
	X		X		X		1946	23849	22649	81	73
		X	X		X		1946	24314	23036	87	88
X			X			X	1946	19343	18731	56	50
	X		X			X	1946	21637	20393	76	64
		X	X			X	1946	23176	22091	90	75
X				X	X		1946	19648	19092	49	51
	X			X	X		1946	24742	23574	72	73
		X		X	X		2052	30135	29490	39	43
X				X		X	1946	19314	18689	49	46
	X			X		X	1946	22120	20850	78	67
		X		X		X	2071	31777	31074	36	38

Table 5: Comparison among Different Combinations of Selection and Tie Breaking Rules and the Changes in Measures of Effectiveness With and Without Exchange Algorithm

Quarter / Year	Number of Platoons	Number of Weeks	Total Number of Periods	Total Runtime	Penalty for Topics Not Scheduled (Total Number of Topics Not Scheduled)	Penalty for Topics Scheduled outside Suggested Timeframe (Total Number of Topics Scheduled outside Suggested Timeframe)				Number of More Than Three Consec. Lectures
						Company controlled	Garrison Training Area only	Firing Range only	External Instructors only	
				[min:sec. sec/100]	[MOE1]	[MOE2]	[MOE2g]	[MOE2f]	[MOE2o]	[MOE3]
I/94	3	13	541	7:14.77	11495 (112)	15675 (105)	8622 (114)	966 (21)	2982 (26)	44
II/94	3	13	544	7:17.30	1946 (53)	18731 (130)	9348 (141)	1259 (21)	3878 (32)	50
III/94	3	13	576	7:17.10	83 (7)	18089 (126)	9990 (132)	1290 (22)	2286 (20)	52
IV/94	3	12	532	7:15.77	2316 (25)	11733 (132)	39147 (192)	2515 (20)	3217 (28)	45

Table 6 : Calculation for Year 1994 - Accumulated for All Platoons

Quarter / Year	Number of Platoon	Number of Weeks	Total Number of Periods	Total Runtime	Penalty for Topics Not Scheduled (Total Number of Topics Not Scheduled)	Penalty for Topics Scheduled outside Suggested Timeframe (Total Number of Topics Scheduled outside Suggested Timeframe)				Number of More Than Three Consec. Lectures
						Company controlled	Garrison Training Area only	Firing Range only	External Instructors only	
				[min:sec. sec/100]	[MOE1]	[MOE2]	[MOE2g]	[MOE2f]	[MOE2o]	[MOE3]
II/94	all	13	544	7:17.30	1946 (53)	18731 (130)	9348 (141)	1259 (21)	3878 (32)	50
	1.	13	544		517 (19)	6104 (38)	3116 (47)	418 (7)	1139 . (11)	15
	2.	13	544		813 (16)	6094 (41)	3116 (47)	324 (7)	1307 (12)	19
	3.	13	544		616 (18)	6533 (51)	3116 (47)	517 (7)	1432 (9)	16

Table 7 : Schedule Calculation for Second Quarter 1994 - Comparison among Platoons

B. SCENARIO OF USE

With this program the task of producing a schedule is simplified and the amount of time needed to complete it drastically reduced. This section describes a scenario of use to illustrate how the scheduling with this computer-aided approach might be done. The times for the tasks are estimates. Field tests to verify the numbers were not performed.

For this scenario it is assumed that the scheduler is not new on the job; he has already done at least one schedule and is familiar with the present manual way of doing it. Furthermore, it is assumed that all the data have been provided to the company and the scheduler with the guidelines the company commander wants to have followed. The time he begins working on the schedule is assumed to be one morning.

At first before actually starting the work the scheduler sits down and evaluates the situation. He compares the framework for this quarter with the last one or if he can with the same quarter last year. He should determine whether the number of platoons is going to change or major other issues have to be incorporated. This should take about 15 to 30 minutes. After this he should collect and sort all his papers and orders with given dates and times relating to the quarter to have them ready to supply the data to the machine. A well organized Sergeant should take at most 15 minutes for this step including the check for completeness.

After this he can start entering the data into the machine. Assuming that he is not that familiar with a keyboard and has to look at it to do his typing he might enter a set of numbers, for example for a day on a Firing Range, in 30 seconds. He should take the same time for other sets of data. Using the sample data (Appendix G) as a reference this would take approximately 40 minutes. Now the user should review the data in the machine for completeness and correctness. This could take up to 20 minutes.

He has now spent between 75 and 90 minutes to prepare the machine and himself for a schedule. The program can be started now. In approximately ten minutes he has the computer calculated schedule he can work with.

The result can be printed or evaluated on the computer screen. For better comparison it might be good to use the printout because limited screen size makes it impossible to display platoons or weeks simultaneously. The time for printout depends strongly on the printer (9- or 24-pin dot matrix). Assuming a worst case this may take half an hour. Now he can evaluate the schedule for the remaining hours of the morning and

implement his changes. This gives him three hours to work on this. The reprint for review by the company commander and the platoon leaders can be done during the lunch break because the printing needs no supervision.

That afternoon, each of the platoon leaders takes time to review his schedule and to suggests changes. Simultaneously the company commander reviews all schedules and demand changes. This might take three hours and in the last hour a decision conference could be held to coordinate suggested changes. One day has been spent on the production of a new schedule.

In the morning of the next day the Sergeant could implement the changes decided the previous afternoon and have the revised schedule ready early that morning. With printing multiple copies he might be finished with the schedule at lunch time. In this scenario with the underlying assumption that the personnel involved are available to review, the schedule is completed in 1.5 days instead of the current 10 to 15 working days of manual labor with paper and pencil or index cards on a huge billboard.

C. BENEFITS AND ACCEPTABILITY

Using a computer-aided approach to scheduling can reduce the time needed to complete a schedule. Furthermore, the schedule can be checked for consistency and violations of any prerequisite chains. The observance of all requirements can be tested quickly. Reducing the time needed to produce the result should free the Sergeant for his other tasks and thus increase the level of readiness of the company in the area of his primary responsibility.

The acceptability of the program has to be evaluated at two levels. The first level is the user level and the second is superior command level. The user will evaluate the program from the usability point of view. The superior command will focus on conformity with regulations.

The main criteria for acceptance on the user level are ease of use and the produced result. Due to the structure of the program with its graphical user interface the program is easy to use and the untrained scheduler new to the job can early on start using it. The data format of date and time entries places the scheduler on a familiar basis with the data he is dealing with. The result produced is delivered in a short time and the schedule produced provides a good basis for the training and possible user interaction and improvements.

Therefore, after getting used to having a computer calculated schedule the gain in time will support the acceptance of the program.

For superior command the underlying algorithm and field testing results are more important. Any decision aid that is introduced into the Armed Forces has to conform to regulations. This algorithm tries to emulate the regulations and guidelines from the catalog of training objectives as close as possible and minimizes any diversions from those rules. Those diversions are introduced not by the program but by local conditions which could not be changed by the company. Combined with the additional advantage of reducing the time to produce the result this program should be acceptable by superior commands after extensive field testing.

D. TRAINING REQUIREMENTS

Since this program has been constructed with an easy to use graphical user interface that features messages and labels that specify what has to be written where, the training to use this program can be minimized. A tutorial with a sample data set that provides the necessary data to perform the calculation should be sufficient for a new scheduler to train himself. The sample data set would allow a calculation that then could be modified to test the exchange possibilities and their messages. The data set would also allow for change in the data to do a different calculation. This data set could be one similar to the set that was used in this paper to determine the best combination of rules for the algorithm (Appendix G).

If the new scheduler is already computer literate it should take no longer than a morning or afternoon to get familiar with the principle use of the program. A computer illiterate person will need more time to learn basics about the machine first.

Although there is no need for a special training course, incorporating this program into the course a NBC-Defense-Sergeant has to take to learn the tasks for NBC-Defense might be appropriate. This then insures that he knows what is available and may be more willing to use it when he is confronted with it the first time he takes the job. It also prevents misinformation about the program being passed from a scheduler to his successor.

VI. CONCLUSIONS AND RECOMMENDATIONS

This thesis has shown that computer-aided scheduling of basic training objectives is possible. The heuristically constructed schedule forms a solid basis on which manual changes can be made. The resulting schedule has face validity and could be used to execute training in the form given. The program provides a quick solution with a calculation time less than the usual morning coffee break instead of 10 to 15 working days of manual labor. In addition to the raw calculation time, the time for data input has to be considered. This time depends on the typing skill of the user, but a complete set of data should rarely take more than one hour to enter.

Therefore, the goal to give the NBC-Sergeant more time to perform his primary duties can be achieved with this program. The options for user interaction were created to allow the user to modify and update the calculated schedule to reflect changes. The correct use of these options can prevent a drift between actual implementation and the data for the schedule stored in the computer. This requires that every desired change is not only made on paper or in training itself but has to be entered into the computer. If this is handled properly, the schedule in the computer can always reflect the training of the platoons.

This program supplies the user with a schedule in code number format and does not provide facilities to print any other form of the schedule. Due to the variety of requirements on how a schedule has to look either by battalion standards or division standards, it was not considered appropriate to prescribe a format and set a standard that might not be acceptable.

Further areas of continuing the research might be in modifying the improvement algorithm to make it more sophisticated. This might increase the number of topics that are scheduled in accordance with the suggested time frame. Another step in the direction of improving the schedule could be implementing the backtracking option which is supplied as an empty procedure in the current version. This might create a better initial solution. Having shown that computer-aided scheduling of a large number of topics into a large number of periods is possible in a reasonable time, another step can be made to consider scheduling for the non-recruit catalogs of training objectives as well. This would certainly increase the magnitude of the problem because some of those have a time span of nine months instead of three months. But those longer scheduling tasks are coupled with more suggestions and restrictions that simplify the manual scheduling process for the soldier.

APPENDIX A: A MONTHLY VIEW OF MAIN SCHEDULE OUTPUTS

April 1994

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
				1	2	3
4	5 Arrival of new Recruits	6	7	8	9 Duty	10
11	12 Garrison Training Area D	13	14	15	16	17
18	19 Garrison Training Area D	20	21 Firing Ranges 2A D	22	23	24
25 Firing Ranges 1A D	26	27	28 Garrison Training Area D	29	30	

Figure A1: Calendar for April 1994 With Major Assignments

APPENDIX B: PARTIAL PRINTOUT OF A SCHEDULE

This appendix includes two weeks as they would appear in a paper printout. The first is week 1 for platoon 1. The second is week 5 for the same platoon. The complete printout of this quarterly schedule requires 39 pages.

```
Week :  1                                                                   Platoon: 1
```

Time	Monday	Tuesday	Wednesday 06/04	Thursday 07/04	Friday 08/04	Saturday 09/04
0700-0745			990000309	010101402	090102802	010106801
0750-0835			990000309	010101403	090102802	090102802
0905-0950			990000309	100101500	090102802	100101500
0955-1040			990000309	100101500	090102802	100103000
1045-1130			990000309	140101001	090102302	110101100
1230-1315			990000309	140101002	090102802	
1320-1405			990000309	010209001	160103901	
1435-1520			990000309	090102801	160103902	
1525-1610			010101401	090102802	160103903	
1615-1700				090102802		

Figure B1: Schedule for Platoon 1, Week 1

Week : 5 Platoon: 1

Time	Monday 02/05	Tuesday 03/05	Wednesday 04/05	Thursday 05/05	Friday 06/05	Saturday
0000-0100			010209003			
0100-0200			010107002			
0200-0300			010107002			
0300-0400			010107002			
0400-0500			010108302			
0500-0600			010111800			
0600-0700			010111800			
0700-0745	100105602	990000306	010201902	090105600	110301000	
0750-0835	100105602	010110500	010201902	090105600	060104403	
0905-0950	090201000	010110500	010202102	090105600	010116101	
0955-1040	090201000	010110500	010202103	090105600	110301000	
1045-1130	060104401	010111800	010202103	090105600	010307002	
1230-1315	060104401	010116102	010204700	090105600		
1320-1405	060104402	010116102	010204700	090202200		
1435-1520	010307002	010116103	010204700	090202200		
1525-1610	990000304	010116103	990000308	090202200		
1615-1700		010116104		060104403		
1700-1800		010201901				
1800-1900		010201901				
1900-2000		010202101				
2000-2100		010209003				
2100-2200		010209003				
2200-2300		010209003				
2300-2400		010209003				

Figure B2: Schedule for Platoon 1, Week 5

APPENDIX C: PRINTOUT OF A REPORT

This report is produced by the scheduling program. The run was performed using the test data set listed in Appendix G. The margins have been modified to fit the page.

```
Date:          4/ 1/1995
Starttime:  17:15:50:38

Platoons This Quarter    :     3
Total Number Of Weeks    :    13
Total Number Of Periods  :   544
Total Number Of Topics   :   402
Number Of Required Hours :   546

Intermediate Result After Garrison Training Area :

Already assigned              : 168 Hours     30.88%
Remaining to be assigned      : 399 Hours
New Number Of Required Hours  : 567 Hours

Intermediate Result after Firing Ranges :

Already scheduled for platoon 1             : 226 Hours      41.54%
Already scheduled for platoon 2             : 226 Hours      41.54%
Already scheduled for platoon 3             : 226 Hours      41.54%
Remaining to be scheduled for platoon 1     : 345 Hours
Remaining to be scheduled for platoon 2     : 344 Hours
Remaining to be scheduled for platoon 3     : 347 Hours
Updated Number of hours needed for platoon 1 : 571 Hours
Updated Number of hours needed for platoon 2 : 570 Hours
Updated Number of hours needed for platoon 3 : 573 Hours

Results after resolving possible conflicts after prescheduling :

Now already scheduled for platoon 1              : 230 Hours      42.28%
Now already scheduled for platoon 2              : 230 Hours      42.28%
Now already scheduled for platoon 3              : 234 Hours      43.01%

          Total      Platoon 1  Platoon 2  Platoon 3
MOE 1 :  90170(824) 30248(275) 30247(274) 29675(275)
MOE 2 :    142( 14)   376( 4)    279( 4)    487( 6)
MOE 2g:   9348(141)  3116(47)   3116(47)   3116(47) Garrison Training
MOE 2f:   1259( 21)   418( 7)    324( 7)    517( 7) Firing Ranges
MOE 2o:      0(  0)     0( 0)      0( 0)      0( 0) External Personnel
MOE 3 :      0          0          0          0

Algorithm Start: 17:15:53:46

Algorithm End:   17:21:44:60
```

```
Total      Platoon 1   Platoon 2   Platoon 3
MOE 1 :   1946( 53)    517(19)     813(16)      616(18)
MOE 2 : 19343(135)    6318(39)    6204(42)     6821(54)
MOE 2g:   9348(141)   3116(47)    3116(47)     3116(47) Garrison Training
MOE 2f:   1259( 21)    418( 7)     324( 7)      517( 7) Firing Ranges
MOE 2o:   3878( 32)   1139(11)    1307(12)     1432( 9) External Personnel
MOE 3 :     56          16          19           21

MOE after exchanging scheduled hours:

            Total     Platoon 1    Platoon 2    Platoon 3
MOE 1 :   1946( 53)    517(19)      813(16)      616(18)
MOE 2 : 18731(130)    6104(38)     6094(41)     6533(51)
MOE 2g:   9348(141)   3116(47)     3116(47)     3116(47) Garrison Training
MOE 2f:   1259( 21)    418( 7)      324( 7)      517( 7) Firing Ranges
MOE 2o:   3878( 32)   1139(11)     1307(12)     1432( 9) External Personnel
MOE 3 :     50          15           19           16

The following topics could not be scheduled:
 Platoon 1        Platoon 2        Platoon 3
060103105-00     060103105-00
                 100301400-01     100301400-01
100301400-02     100301400-02     100301400-02
990000312-09
990000312-10                      990000312-10
990000312-11                      990000312-11
990000312-12                      990000312-12
990000312-13     990000312-13     990000312-13
990000312-14     990000312-14     990000312-14
990000312-15     990000312-15     990000312-15
990000313-00     990000313-00     990000313-00
990000314-01     990000314-01     990000314-01
990000314-02     990000314-02     990000314-02
990000314-03     990000314-03     990000314-03
990000314-04     990000314-04     990000314-04
990000314-05     990000314-05     990000314-05
990000314-06     990000314-06     990000314-06
990000314-07     990000314-07     990000314-07
990000314-08     990000314-08     990000314-08
990000314-09     990000314-09     990000314-09

Date:        4/ 1/1995
Starttime: 17:23:10:67
```

APPENDIX D: DISPLAY OF SUB-MENUS IN THE INTERFACE

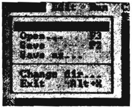

Figure D1: 'File' Menu

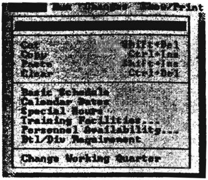

Figure D2: 'Edit' Menu

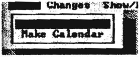

Figure D3: 'Run' Menu

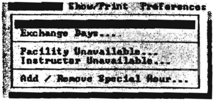

Figure D4: 'Change' Menu

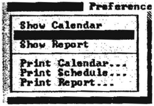

Figure D5: 'Show / Print' Menu

Figure D6 : 'Preferences' Menu

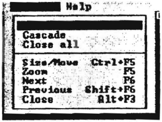

Figure D7: 'Window' Menu

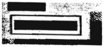

Figure D8 : 'Help' Menu

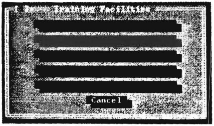

Figure D9: Facility Selection Menu

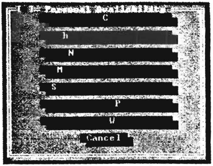

Figure D10: Instructor Selection Menu

APPENDIX E: DISPLAYS OF INPUT WINDOWS

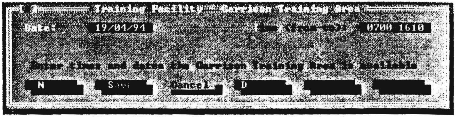

Figure E1: Dialog Box for Garrison Training Area

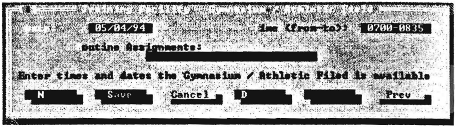

Figure E2: Dialog Box for Gymnasium / Athletic Field

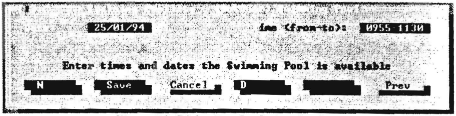

Figure E3: Dialog Box for Swimming Pool

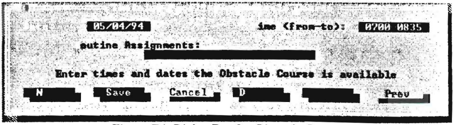

Figure E4: Dialog Box for Obstacle Course

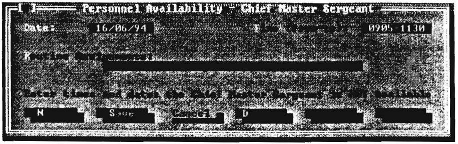

Figure E5: Dialog Box for Chief Master Sergeant Unavailability

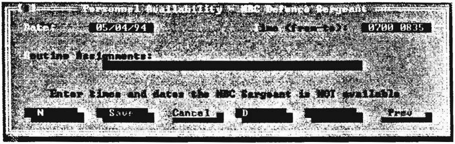

Figure E6: Dialog Box for Nuclear-Biological-Chemical-Defense Sergeant Unavailability

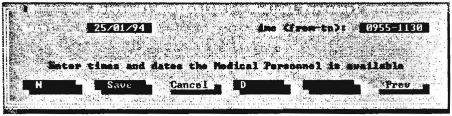

Figure E7: Dialog Box for Medical Personnel Availability

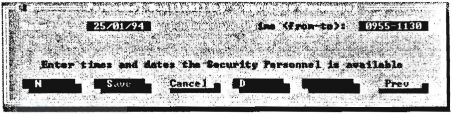

Figure E8: Dialog Box for Security Personnel Availability

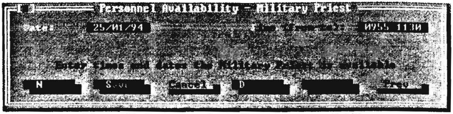

Figure E9: Dialog Box for Military Priest Availability

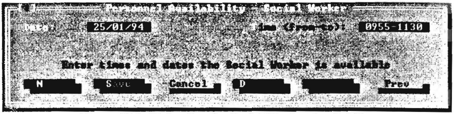

Figure E10: Dialog Box for Social Worker Availability

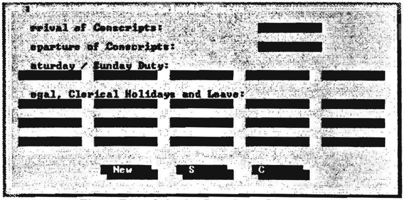

Figure E11: Calendar Data Input Dialog Box

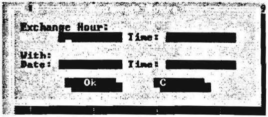

Figure E12: Dialog Box for Interchanging Hours

Figure E13: Dialog Box for Interchanging Days

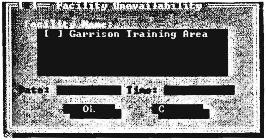

Figure E14: Dialog Box to Handle Unavailability of a Facility

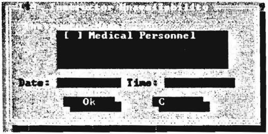

Figure E15: Dialog Box to Handle Unavailability of an External Instructor

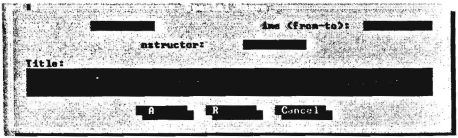

Figure E16: Dialog Box to Add or Remove a Special Hour or a Special Event

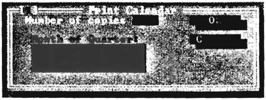

Figure E17: Dialog Box to Select Calendar Printing

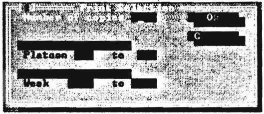

Figure E18: Dialog Box to Select Schedule Printing

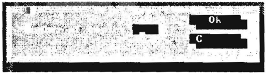

Figure E19: Dialog Box to Determine Number of the Copies to Print

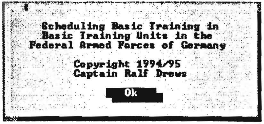

Figure E20: About Box

APPENDIX F: DISPLAYS OF WARNINGS AND MESSAGES

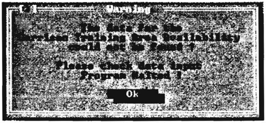

Figure F1: Warning Box for Missing Garrison Training Area Data

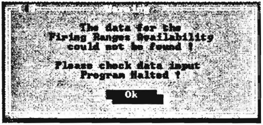

Figure F2: Warning Box for Missing Firing Range Data

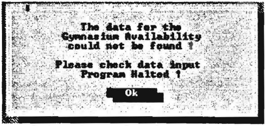

Figure F3: Warning Box for Missing Gymnasium Data

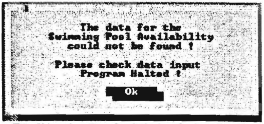

Figure F4: Warning Box for Missing Swimming Pool Data

Figure F5: Warning Box for Missing Obstacle Course Data

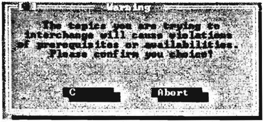

Figure F6: Confirmation Request for Exchange with Constraint Violation

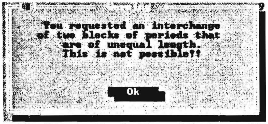

Figure F7: Warning Box for Unequal Exchange of Hours

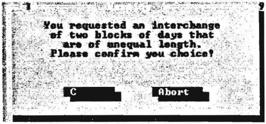

Figure F8: Confirmation Request for Unequal Exchange of Days

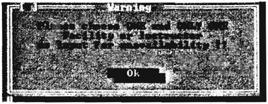

Figure F9: Error Message for More Than One Check Mark

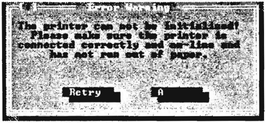

Figure F10: Printer Error Warning

APPENDIX G: SAMPLE INPUT DATA FOR ONE QUARTER

This appendix features the quantity of input data needed for the program to produce a schedule. Data for instructor availability or unavailability can be omitted. As described, the program assumes that if they are omitted, they are either never or always available. The data for facilities have to be provided to produce a meaningful schedule. The algorithm does not execute if these data are missing. All the sample data refer to the second quarter of 1994. The date entries in the tables are given in the format Day/Month/Year. Time data are given in the 24 hour format.

Event	Date
Arrival of Recruits	05/04/94
Departure of Recruits	30/06/94
Duty on Saturday	08/04/94
Legal / Clerical Holiday	12/05/94
Legal / Clerical Holiday	13/05/94
Legal / Clerical Holiday	23/05/94
Legal / Clerical Holiday	02/06/94
Legal / Clerical Holiday	03/06/94

Table G1: Calendar Dates for Second Quarter 1994

Date	Time	Instructor	Title / Topic
14/04/94	1530-1610	CpCdr	Promotions and Awards, Company Muster
16/05/94	0700-0835	BtlCdr	Information on Actual Political Developments
27/06/94	0700-0835	BtlCdr	Quarterly Battalion Muster

Table G2: Special Events / Hours Not Covered in Catalog of Training Topics

Date	Time
12/04/94	0700-1610
19/04/94	0700-1610
28/04/94	0700-1700
03/05/94	0700-2400
04/05/94	0000-1610
10/05/94	0700-1610
17/05/94	0700-2400
18/05/94	0000-1610
31/05/94	0700-1610
06/06/94	0700-1610
07/06/94	0700-1610
21/06/94	0700-2400
22/06/94	0000-1610

Table G3: Garrison Training Area Availability in Second Quarter 1994

Date	Time	Number of Subranges A	Number of Subranges D
21/04/94	0700-1610	2	0
25/04/94	0700-1610	1	0
05/05/94	0700-1700	2	2
11/05/94	0700-2400	3	0
25/05/94	1230-2400	2	1
09/06/94	0700-1700	2	1

Table G4: Firing Ranges for the Company

Day of Week	Starting Date	Time	Routine Assignment
Friday	08/04/94	0955-1130	weekly

Table G5: Obstacle Course Assignments

Day of Week	Starting Date	Time	Routine Assignment
Thursday	07/04/94	0905-1040	weekly
Friday	08/04/94	0955-1130	weekly
Saturday	09/04/94	0700-1130	--
Monday	11/04/94	0700-0835	weekly
Monday	11/04/94	1435-1610	weekly
Wednesday	13/04/94	0700-0835	weekly

Table G6: Gymnasium / Athletic Field Assignments

Date	Time
13/04/94	0700-0835
26/04/94	0955-1130
02/05/94	0700-0835
24/05/94	0955-1130
24/05/94	1230-1405
24/05/94	1435-1610
14/06/94	0955-1130
28/06/94	0700-0835
28/06/94	1230-1405

Table G7: Outdoor / Indoor Swimming Pool Assignments

Function (Billet)	Day of Week	Starting Date	Time	Routine Assignment
CpCdr	Monday	04/04/94	0700-0835	--
CpCdr	Tuesday	05/04/94	0700-0835	--
CpCdr	Wednesday	06/04/94	0900-1130	weekly
CpCdr	Wednesday	15/06/94	0700-1610	--
CpCdr	Thursday	16/06/94	0905-1130	--
CMSgt	Wednesday	15/06/94	0700-1610	--
CMSgt	Thursday	16/06/94	0905-1130	--
CMSgt	Monday	20/06/94	0700-1610	daily
NBCSgt	Monday	04/04/94	0700-0835	--
NBCSgt	Tuesday	05/04/94	0700-0835	--
NBCSgt	Wednesday	06/04/94	0900-1130	--
NBCSgt	Wednesday	15/06/94	0700-1610	--
NBCSgt	Thursday	16/06/94	0905-1130	--

Table G8: Unavailability Data for Company Instructors

Function	Date	Time
MedPers	18/04/94	0700-0835
MedPers	20/04/94	1230-1405
MedPers	21/04/94	0700-0835
MedPers	25/04/94	0955-1130
MedPers	29/04/94	0700-0835
MedPers	06/05/94	0900-1130
SecPers	25/04/94	0955-1130
SecPers	26/04/94	1435-1610
SecPers	29/04/94	0700-0835
SecPers	09/05/94	0900-1130
SecPers	16/05/94	0700-0835
MilPriest	22/04/94	0905-1040
MilPriest	20/05/94	0750-0950
MilPriest	24/06/94	0750-0950
SocWork	15/04/94	0955-1130
SocWork	27/05/94	0900-1040

Table G9: Availability of External Instructors

Numeric Code of Topic	Requirement
160101304	only on Friday a.m.
1001X	at least two times a week
1101X	at most two times a week

Table G10: Additional Battalion / Division Requirements

APPENDIX H: PSEUDOCODE FOR PROGRAM

Pseudocode for coding date and time data into periods:

```
read all input data for calendar dates;
read all input data from basic schedule matrix;
read availability data for garrison training area and firing ranges;
CurrentDay ← arrival date + 1;
PeriodCounter ← 1;
while not at the date recruits leave do
  if it is not a holiday then
    determine which day of the week is CurrentDay;
    if it is not a Saturday or Sunday then
      while periods in basic schedule not empty do
        code period with date, time and day of week information;
        code period with daylight information as 'Day';
        increment PeriodCounter;
    else
      while periods on Wednesday afternoon in basic schedule not empty do
        code period with date, time and day of week information of previous day;
        code period with daylight information as 'Day';
        increment PeriodCounter;
      while periods Friday in basic schedule not empty do
        code period with date, time and day of week information;
        code period with daylight information as 'Day';
        increment PeriodCounter;
  ;
  increment CurrentDay;
;
for all dates and times in garrsion training area list and firing range list do
  find the day in period list;
  compare the time with first and last period times on that day;
  if they do not match then
    calculate how many more periods on that day are needed;
    determine whether early or late on that day;
    find the insertion point;
    move all later topics for the amount to insert;
    increase the period counter;
    insert the periods with date and new time information;
    code the day of week information same as the day used;
    code the daylight information depending on the time and the quarter;
;
save period list and PeriodCounter;
```

Pseudocode for interchange hour algorithm:

```
read data from special log file;
read the date and time information from the dialog box;
determine platoon from opened schedule;
determine status from override value;
convert dates and times into start periods and durations;
if durations are not equal then
  show warning box and abort;  {this cannot be executed}
else
  check for violations;
  if no violations or user override then
    for j := 0 to duration - 1 do
      Temp ← Schedule[platoon, start period1 + j];
      Schedule[platoon, startperiod1 + j] ← Schedule[platoon, start period2 + j];
      Schedule[platoon, start period2 + j] ← Temp;
    ;
  else
     display confirmation box;
;
```

Pseudocode for interchange day algorithm:

```
read data from special log file;
read the date and time information from the dialog box;
determine platoon from opened schedule;
determine status from override value;
convert dates and times into start periods and durations;
if durations are not equal then
  if not override then
    show warning box and abort;
  else
    Length ← Minimum(duration1 and duration2);
    for j:= 0 to Minimum(duration1 and duration2) do
       Temp ← Schedule[platoon, start period1 + j];
       Schedule[platoon, start period1 + j] ← Schedule[platoon, start period2 + j];
       Schedule[platoon, start period2 + j] ← Temp;
    ;
    if duration1 < duration2 then
      first ← start period1;   {first always holds day that was shorter}
      second ← start period2;
    else
      first ← start period2;
      second ← start period1;
```

```
      for the shorter day find the periods that do not match;
      insert as many as needed into the period list;
      for all platoons do
         shift the schedule for the number of inserted periods;
         schedule a '0' {not existing topic = unscheduled} there;
      ;
      for j := 1 to Abs(duration1 - duration2) do
         Schedule[platoon, first + Length + j] ←
                        Schedule[platoon, second + Length + j];
         Schedule[platoon, second + Length + j] ← 0;
   ;
else
   check for violations;
   if no violations or user override then
      for j := 0 to duration - 1 do
         Temp ← Schedule[platoon, start period1 + j];
         Schedule[platoon, start period1 + j] ← Schedule[platoon, start period2 + j];
         Schedule[platoon, start period2 + j] ← Temp;
      ;
   else
      display confirmation box;
;
```

Pseudocode for handling instructor unavailability:

```
read name of instructor from dialog box;
read date and time instructor is no longer available;
convert date and time into start period and duration;
if instructor = instructor that teaches each platoon individually then
   Platoon ← platoon number from the displayed schedule;
   for j := 0 to (duration - 1) do
      unschedule Schedule[Platoon, start period + j];
   ;
   while an unscheduled period exists do
      search the list of topics for a topic on a facility that is available and not used
      by other platoons and the length of topic ≤ remaining unscheduled periods;
      if such a topic found then
         schedule it;
      else
         search for an unscheduled topic not in group 99 and length ≤ remaining
         unscheduled periods;
         if (such a topic exists) and (instructor is available) then
            schedule it;
         else
```

```
            schedule topic from group 99;
         ;
      ;
   ; {end while}
else
   for plt := 1 to total number of platoons do
      for j := 0 to (duration - 1) do
         unschedule Schedule[plt, start period + j];
      ;
      while an unscheduled period exists do
         search the list of topics for a topic on a facility that is available and not used
         by other platoons and the length of topic ≤ remaining unscheduled periods;
         if such a topic found then
            schedule it;
         else
            search for an unscheduled topic not in group 99 and length ≤ remaining
            unscheduled periods;
            if (such a topic exists) and (instructor is available) then
               schedule it;
            else
               schedule topic from group 99;
            ;
         ;
      ; {end while}
   ;
;
update schedule display;
save schedule in special log file;
;
```

<u>Pseudocode for adding a special hour / event:</u>

```
read date and time from dialog box;
convert date and time into period and duration;
count number of special events /hours in list of topics;
append new topic to list of topics;
name it 'Special' + (Number of special event /hours + 1);
increase the total number of topics;
Platoon ← platoon number from the displayed schedule;
if scheduled topic at period is not part of a chain then
   unschedule it;
   for j := 1 to duration do
      Schedule[Platoon, period + j - 1] ← index to special topic;
   ;
else
```

```
    save topics that have to be replaced into set of replacements;
    for j := 1 to duration do
       Schedule[Platoon, period + j - 1] ← index to special topic;
    ;
    while set of replacements not empty do
       shift replaced topics to next earliest scheduling opportunity;
       if (replaced topic can be scheduled prior to its decendant) and
            (scheduled topic in group 99) then
          schedule it;
          remove topic from set of replacements;
          unschedule the other topic there;
       else
          append topics to set of replacements;
          schedule first topic in set of replacements and remove it from set;
          ;
    ; {end while}
;
update schedule display;
save scheduling information into special log file;
;
```

Pseudocode for removing a special hour / event:

```
read date and time information from dialog box;
Platoon ← number of platoon from displayed schedule;
convert date and time into period and duration information;
for j := 0 to (duration - 1) do
   Schedule[Platoon, period + j] ← 0;
   flag topic as unscheduled;
;
while Schedule has unscheduled hour do
      search the list of topics for a topic on a facility that is available and not used
      by other platoons and the length of topic ≤ remaining unscheduled periods;
    if such a topic found then
       schedule it;
    else
       search for an unscheduled topic not in group 99 and length ≤ remaining
       unscheduled periods;
       if (such a topic exists) and (instructor is available) then
          schedule it;
       else
          schedule topic from group 99;
       ;
     ;
  ; {end while}
```

APPENDIX I: PROGRAM DESCRIPTION

The code is programmed in BORLAND TURBO PASCAL Version 7.0 (Borland, TP, 1992) with extensive use of TURBO VISION that comes with the language. Formats of data the scheduler normally uses are difficult to handle for a computer. Therefore, the data have to be converted or coded for easier reference and use. This section deals with the specific data structures used and implemented in the computer and the data conversion.

The complete listing of the code is available upon request from the institution that approved the thesis or from the author.

1. Data Structures

All data are stored in files and can be retrieved when additional information becomes available or entered information has to be changed. Separate files are kept for the current quarter and the next quarter. Files belonging to the current quarter have as their extension ".qtr" and files belonging to the new quarter ".new". Backup files for the last quarter carry the extension ".old". Data files with other extensions are shared files. The files can be read with a normal text editor but they contain hidden characters that are vital for the functionality of the program. Therefore, the only way to change information in those files properly is via the program. Any attempt to do so using a text editor might result in shifting or even overwriting some of the hidden characters. Those hidden characters are identifiers for the parts of the program that manage the data transfer between a dialog box, the random access memory (RAM) and the file. Modifying those hidden characters make the programs unable to read and to recover the data stored in those files.

2. Data Preparation and Conversion

Prior to initiating scheduling the data have to be converted and coded into an easy accessible and maintainable format as indicated above. A pseudocode description is given in Appendix H. The training objectives in the CTO are stored in an array of records and in the program only referenced by their position in the array. Topics out of the CTO that have to be excluded from scheduling are deleted from the array.

a. Periods

The program uses the notion of periods for assigning topics. Periods are individual training segments of a length between 45 and 60 minutes. All training objectives have to be scheduled in periods in the given training time between the arrival and the departure of the new recruits. An easy maintainable way for the computer to handle the periods is the use of total enumeration. By using the data from the basic schedule of the company in combination with the calendar data, the available periods can be enumerated throughout the quarter. The list of all periods in the quarter is stored as an array of records where the index refers to the number of the period and the fields contain the actual date and actual time. Additionally, the periods are coded with the information on the day of the week and day or night identifier. The first period available for scheduling is the first period on the day following the arrival day, the second period is the next period on that day and the last usable period is the latest period on the day prior to departure. The list of periods contains the daytime periods coded from the base schedule. In addition, periods outside that timeframe from the availability of Garrison Training Areas and Firing Ranges are incorporated.

b. Facilities and Instructors

All data from training facilities, availability of external instructors and non-availability of company instructors are converted into their period equivalent. The use of periods is easier for comparison than using the actual dates and times. Comparing two numbers is less computation effort than comparing strings of data. If some information about instructors has not been given (that is the files are empty), the program assumes that the instructor is either never or always available. External instructors are assumed to be never available and company instructors always. These data are kept in array of linked lists.

c. Other Conversions

The data for the requirements from division and battalion level are coded into a data structure of a single linked list. For special events a combined step of coding and prescheduling is taken. At first they are converted into training objectives and appended to the array of records for the training objectives. Then they are prescheduled into the

appropriate periods in the schedule and marked that they cannot be moved to other periods.

The schedule itself at this point is kept in an [m x n]-two-dimensional matrix, where m is the number of platoons and n the number of periods. Stored in this matrix is only the index of the training objective which has been scheduled to the period n for platoon m.

LIST OF REFERENCES

Borland International, *Turbo Pascal - Programmer's Reference,* 1992, Scotts Valley.

Borland International, *Turbo Vision - Programmer's Guide,* 1992, Scotts Valley.

Heeresamt, *Die Allgemeine Grundausbildung im Heer (AnTrA 1),* 1988, Cologne.

Justice, Barry D., "A Scheduling Model for the U.S. Marine Corps Communication Electronics School," Master's Thesis, Naval Postgraduate School, Monterey, California, September 1993.

Lawrie, N. L., "An Integer Linear Programming Model of A School Timetabling Problem," *The Computer Journal,* vol 12, 1969, pp. 307 - 316.

Leighton, Frank T., " A Graph Coloring Algorithm For Large Scheduling Problems," *Journal of Research of the National Bureau of Standards,* vol. 84, 1979, pp. 489 - 506.

Mulvey, John M., "A Classsroom/Time Assignment Model," *European Journal of Operations Research,* vol 9, 1982, pp. 64 - 70.

Oakford, Robert V. et al, "School Scheduling - Practice and Theory," *Journal Educational Data Processing,* vol 4, 1966, pp. 16 - 50.

Romero, Bernado P., "Examination Scheduling In A Large Engeneering School: A Computer-Assisted Participative Procedure," *Interfaces,* vol. 12, 1982, pp. 17 - 24.

Schmidt, G. and Ströhlien, T., "Timetable Construction - An Annotated Bibliography," *The Computer Journal,* vol 23, 1979, pp. 307 - 316.

Watson, Blake, *Programming with Turbo Pascal,* 1994, New York.

de Werra, D., "An Introduction To Timetabling," *European Journal of Operations Research,* vol 19, 1985, pp. 151 - 162.

Wood, D. C., " A System For Computing University Examination Timetables," *The Computer Journal,* vol. 11, 1968, pp. 41 - 47.

Wood, D. C., "A Technique For Colouring A Graph Applicable To Large Scale Timetabling Problems," *The Computer Journal,* vol. 12, 1969, pp. 317 - 319.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5101	2
3.	Prof. Gordon H. Bradley, Code OR/Bz Department of Operations Research Naval Postgraduate School Monterey, CA 93943-5002	3
4.	Prof. Robert F. Dell, Code OR/De Department of Operations Research Naval Postgraduate School Monterey, CA 93943-5002	2
5.	United States Army Recruiting Command Code N7 Fort Knox, KY 40121	1
6.	Hauptmann Ralf Drews 1. / Panzeraufklärungsbatallion 13 Friedenstein - Kaserne 99867 Gotha Germany	2
7.	German Military Command USA/Canada Attn: S1 StOffz 11150 Sunrise Valley Drive Reston, VA 22091	2